

A Taxonomy of Network and Computer Attack Methodologies

November 7, 2003

Simon Hansman

slh45@cosc.canterbury.ac.nz

Department of Computer Science and Software Engineering
University of Canterbury, Christchurch, New Zealand

Supervisor: Associate Professor Ray Hunt

E : Krys

O : Makund

Abstract

Since the invention of computers and networks, people have found various ways to attack them. Attacks over the years have ranged from using a sledge hammer on a computer, to advanced distributed denial of service attacks. This research focuses on computer and network attacks and providing a taxonomy of them. This is to help combat new attacks, improve computer and network security and to provide consistency in language when describing attacks. A wide range of computer and network attacks are examined to provide both a survey of the field and to provide a basis on which to build the proposed taxonomy. The proposed taxonomy consists of four dimensions which provide a holistic taxonomy and to deal with inherent problems in the computer and network attack field. The first dimension covers the attack vector and the main behaviour of the attack. The second dimension allows for classification of the attack targets. Vulnerabilities are classified in the third dimension and payloads in the fourth. The taxonomy is briefly evaluated and is found to work well, with a few areas that could be improved.

Acknowledgements

Firstly I would like to thank my supervisor, Associate Professor Ray Hunt. This research project has been interesting and I have appreciated having you as a supervisor. Thank you to Jay Garden, Nick Lavery and Michael Clarkson at the Centre for Critical Infrastructure Protection (CCIP) for the ideas and help given through the course of the year. Many thanks to the CCIP and Allied Telesyn for the scholarships. Also to my parents, who have not only lovingly raised me, but who have also helped me greatly in proof reading this report. Thank you to Oliver Hunt for proof reading my report. Finally thank you to all the students in the honours class of 2003. You have made this year enjoyable and interesting.

Contents

1	Introduction	5
2	Background	7
2.1	Computer and Network Attacks	8
2.2	A Brief History of Attacks	9
2.3	Previous Examinations of Attacks	10
2.4	Motivation for Research	10
3	Network and Computer Attack Methodologies	13
3.1	The Attack Process	13
3.2	Viruses	14
3.2.1	File Infectors	14
3.2.2	System and Boot Record Infectors	14
3.2.3	Macro Viruses	14
3.2.4	Virus Properties	14
3.3	Worms	15
3.3.1	Mass-Mailing Worms	15
3.3.2	Network-Aware Worms	15
3.4	Trojans	16
3.4.1	Logic Bombs	17
3.5	Buffer Overflows	17
3.5.1	Stack Buffer Overflow	17
3.5.2	Heap Overflows	17
3.6	Denial of Service Attacks	17
3.6.1	Host Based	18
3.6.2	Network Based	18
3.6.3	Distributed	19
3.7	Network-Based Attacks	19
3.7.1	Spoofing	20
3.7.2	Session Hijacking	20
3.7.3	Wireless Network Attacks	21
3.7.4	Web Application Attacks	21
3.8	Physical Attacks	22
3.8.1	Basic Attacks	23
3.8.2	Energy Weapon Attacks	23
3.8.3	Van Eck Attacks	23
3.9	Password Attacks	23
3.9.1	Password Guessing/Dictionary Attack	23
3.9.2	Brute Force	24
3.9.3	Exploiting the Implementation	24
3.10	Information Gathering Attacks	24
3.10.1	Sniffing	24

3.10.2 Mapping	24
3.10.3 Security Scanning	25
3.11 Blended Attacks	25
4 Toward a Taxonomy	27
4.1 Requirements of a Taxonomy	27
4.2 Existing Taxonomies and Previous Work	28
4.2.1 Early Security Taxonomies	28
4.2.2 Bishop's Vulnerability Taxonomy	28
4.2.3 Howard's Taxonomy	28
4.2.4 Lough's Taxonomy	29
4.2.5 OASIS Web Application Security Technical Committee	30
4.3 The Proposed Taxonomy	30
4.3.1 Overview	31
4.4 Classification	31
4.4.1 The First Dimension	31
4.4.2 The Second Dimension	33
4.4.3 The Third Dimension	34
4.4.4 The Fourth Dimension	36
4.4.5 Other Dimensions	36
5 Evaluation of the Proposed Taxonomy	37
5.1 Introduction	37
5.2 Analysis	37
5.2.1 Requirements	38
5.3 Future Work	39
6 Conclusion	41
A Classifications of Case Studies	43

Chapter 1

Introduction

Security threats to computers and networks have been a problem since computers and networks were first used. Over the past few decades these threats have increased to the point where almost every computer and network is exposed to some form of attack.

The research covered in this paper is split into two main areas. Firstly, a wide range of computer and network attacks are examined and secondly, a taxonomy of attacks is proposed. An attack for the purposes of this research, is an attempt on a computer or network that either damages; discloses information; subverts; or denies or steals services. An attack does not have to be run from a computer, and may be a physical attack as simple as destroying a computer with a sledge hammer. A taxonomy is a systematic way of classifying attacks, so that similar attacks are in the same category. A famous example of a taxonomy is the animal kingdom classification.

There are a number of reasons for examining a wide range of attacks. To combat attacks and to provide a taxonomy it is necessary to understand them. Also previous examinations of attacks have not covered such a wide range. New developments such as the new wave of blended threats and information warfare techniques are examined.

The purpose of the taxonomy is to provide a common means of classifying attacks. Currently attacks are often described differently by different organisations. A taxonomy also allows for previous knowledge to be applied to new attacks and provides a structured way of viewing them.

Chapter 2 examines some of the background issues in the field of network and computer attacks. A brief history is given and previous work is discussed. The motivations for research are also given.

In the following chapter, network and computer attack methodologies are examined. The attack process which most attacks follow is explained first, then a range of computer and network attacks are discussed.

In Chapter 4, the proposed taxonomy is described. Requirements for the taxonomy are proposed from the literature and previous taxonomies are evaluated. The proposed taxonomy is then explained in detail.

Chapter 5 briefly evaluates the proposed taxonomy. The requirements of the taxonomy are re-examined to determine whether the taxonomy met them. Also some weaknesses and advantages in the taxonomy are identified. Finally, future work is discussed.

Chapter 2

Background

In this chapter, some background to the research topic will be discussed. Firstly, a brief look at the problem of computer and network attacks is given. Secondly, computer and network attacks are defined in Section 2.1. In Section 2.2, a brief history of computer and network attacks is given. Section 2.3 discusses some of the previous work done on attack examinations (previous work on attack taxonomies is given in Section 4.2), and finally some of the motivation behind this research is explained.

Since 1999 there has been a marked increase in the number of incidents¹ reported as statistics from the Computer Emergency Response Team Coordination Center[29] (CERT/CC) show. Figure 2.1 shows graphically the number of incidents as reported by CERT/CC over the past eight years. For the first two quarters of 2003 a further 76,404 incidents have been reported.

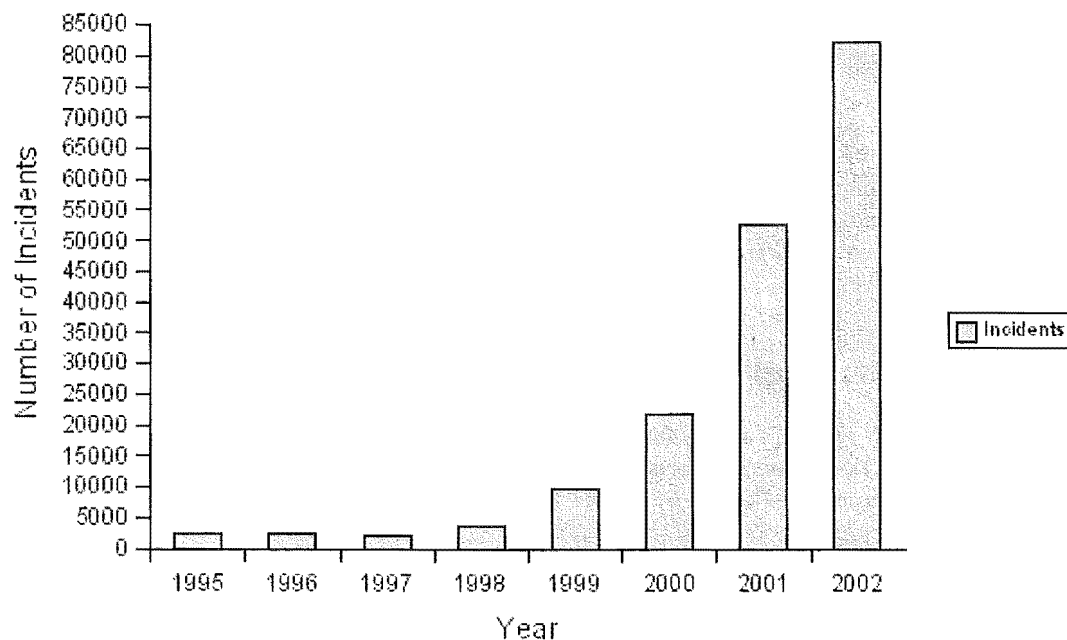


Figure 2.1: Incidents over the Past Eight Years

Not only has there been a marked increase in the number of attacks, the sophistication of the attacks has also increased. With the increased sophistication, many attacks are now relatively “user-friendly” and

¹An incident is an attempt at violating security policy, such as attacking a computer or attempting to gain unauthorised access to some data.

in-depth technical knowledge is no longer required to launch an attack. This has led to the rise of various groups of attackers, such as “script-kiddies”, who while ignorant of how their attack works, can cause great damage. In [46], this trend is represented graphically as shown in Figure 2.2.

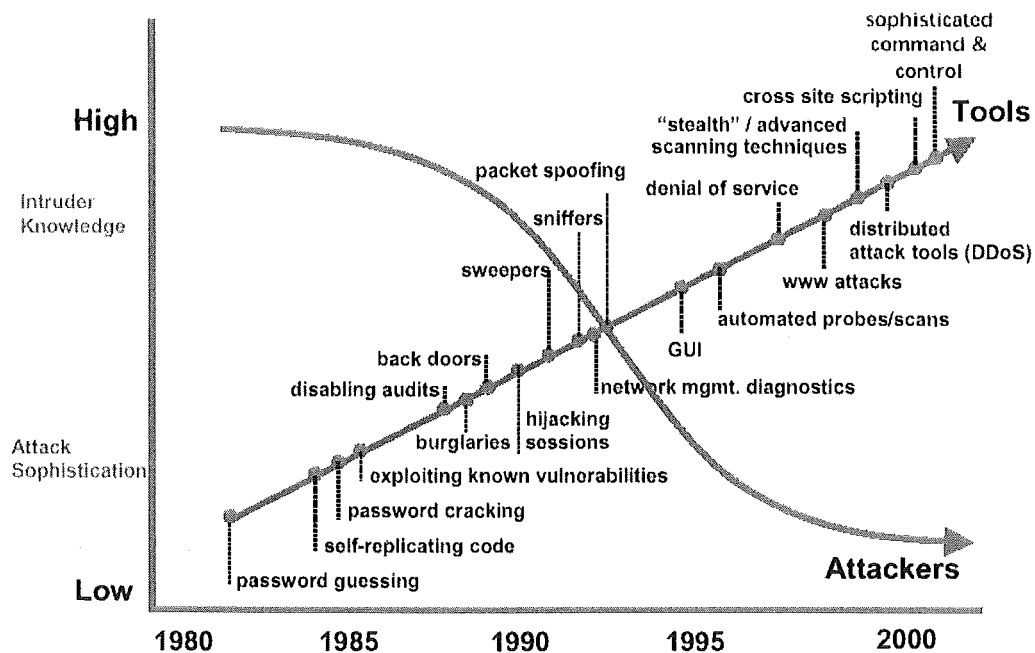


Figure 2.2: Attack Sophistication vs. Intruder Technical Knowledge

Network and computer attacks have become pervasive in today’s world. Any computer connected to the Internet is under threat from viruses, worms and attacks from hackers. Home users, as well as business users, are attacked on a regular basis. Thus the need to combat computer and network attacks is becoming increasingly important.

2.1 Computer and Network Attacks

Before examining the types of attacks that can be launched against a computer or network, it is necessary to explain what network and computer attacks are. Figure 2.3 shows the relationship between them. Network attacks are almost a subset of computer attacks, but some network attacks are outside the computer attack domain.

Computer attacks are attacks aimed at attacking a computer system in some way. This attack may involve destroying or accessing data, subverting the computer or degrading its performance. Traditionally attacks on computers have included methods such as viruses, worms, buffer-overflow exploits and denial of service attacks. These methods, and more, are covered in Chapter 3.

Network attacks are mostly attacks on computers that use a network in some way. A network could be used to send the attack (such as a worm), or it could be the means of attack (such as Distributed Denial of Service attack). An attack on a computer that requires a network, is a network attack. In general, network attacks are a subset of computer attacks.

However, there are several types of network attacks that do not attack computers, but rather the network they are attached to. Flooding a network with packets does not attack an individual computer, but clogs up the network. Although a computer may be used to initiate the attack, both the target and the means of attacking the target are network related.

For the purposes of this research, the term attack (both for network and computer attacks) is broad enough to cover a wide range of attacks, ranging from viruses to physical attacks. The range of attacks is discussed in the next chapter.

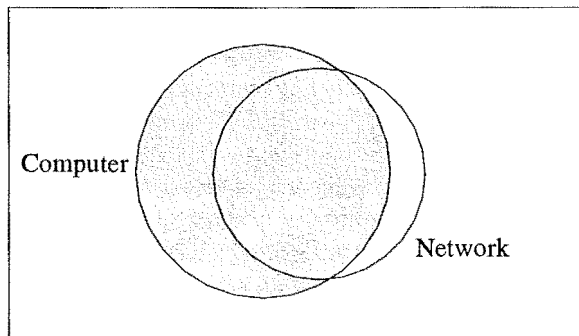


Figure 2.3: The Relationship Between Computer and Network Attacks

2.2 A Brief History of Attacks

Computer and network attacks have evolved greatly over the last few decades. Since computers and networks were invented, there has always been the opportunity to attack them. However, over the last 25 years attacks have split into distinct categories. New attacks, such as worms and viruses have been developed and attacks have become increasingly complicated. Figure 2.2 shows this trend and shows some of the trends in the history of attacks. Some of the more important developments in the history of computer and network attacks are discussed below.

In 1978, the concept of a worm[31] was invented by researchers at the Xerox Palo Alto Research Center. Although the original worm was designed to promote efficient use of computers by searching a network for idle computers, the concept was used by Robert Morris to release the first Internet worm: The Morris Worm[67]. The first viruses were released in 1981, among them Apple Viruses 1, 2 and 3 which targeted the Apple II operating system. In 1983, Fred Cohen was the first person to formally introduce the term “computer virus”² in his thesis[33], which was published in 1985. Over the next decade viruses became more common and prompted the development of anti-virus tools.

More recently, new attacks such as denial of service (DoS) and distributed DoS (DDoS) attacks have been developed. While DoS attacks such as pulling out the power cord of a computer have been around since computers have been invented, new forms using networks and processes on computers were developed in the mid 1990s. DDoS attacks were first seen in 1999 with the introduction of a number of tools to automate the attack (see Section 3.6.3).

Two major recent developments in computer and network attacks are blended attacks and information warfare. Both have influenced the way in which new attacks are being created and both will shape the future of attacks.

In 2001 a new wave of attacks began where existing attack techniques were blended together. The first of the new wave of blended attacks was seen on June 18th 2001 with the release of Code Red[32, 23]. Blended attacks contain two or more attacks merged together to produce a more potent attack. The deadliness of blended attacks soon became apparent, with the damage caused by Code Red, Nimda[24], Slammer[27] and Blaster[28]. Slammer became the fastest worm in history and dramatically reduced network performance across the Internet.

However, blended attacks are not a new form of attack. The original Morris worm[38, 67], released in 1988, was a blended attack. The Morris worm used multiple Unix vulnerabilities to spread. While

²Science fiction author David Gerrold used the term in some of his short stories.

blended attacks have been around since 1988, the new wave of blended attacks are far more damaging and more effective than previous blended attacks. Since Code Red, the number of blended attacks has increased. Symantec's Internet Security Threat Report Volume III[69] released in February 2003 found that blended threats are the greatest risk to the Internet community and that the potential existed for even more damaging blended threats. Months later, Slammer and Blaster wrecked havoc on the Internet. In the following volume[70], Symantec found that blended threats had increased 20% from the last half of 2002.

The new wave of blended attacks is still gathering momentum. As more vulnerabilities are discovered, blended attacks become more common and more damaging. More details on the technical aspects of blended attacks can be found in Section 3.11.

Information warfare is a new and developing area of research. No common consensus has yet been reached on what information warfare is precisely. It is apparent that information warfare is an evolution in the way war is waged. Information warfare is essentially a country using relevant information to attack another country or defend itself. Instead of just waging war with bullets, information is used as a weapon. The attacks used in information warfare are varied. Traditional computer and network attacks are used, as well as less traditional attacks such as Electromagnetic Pulse (EMP) weapons. Some of the electronic means of information warfare are discussed in Section 3.8.

2.3 Previous Examinations of Attacks

There are many previous examinations of computer and network attacks, including both general and specific examinations. General examinations cover a range of attacks, as is done in Chapter 3, while specific examinations analyse one attack in great detail.

Work done by Chris Rodgers[64] covers many computer and network attacks with regards to TCP/IP networking. His research was carried out in 2001 and provides a good overview of the threats and attacks that face TCP/IP networking, as well as attacks such as viruses, worms, trojans and denial of service attacks. The research is still relevant today, however there have been a number of recent developments that need to be examined. Rodgers examines the Code Red worm, which was the first of the new wave of blended attacks, but does not examine the blended nature of the worm. Since Code Red, blended attacks have become one of the major threats, and so an examination of these attacks is needed. Furthermore, Rodgers does not cover a number of attacks such as information warfare attacks and web application attacks.

Other less extensive examinations exist, such as [43]. In [43] a general overview of the types of attacks that are a threat to Internet security is given, as well as analysis of some of the trends attacks are following.

The CERT/CC is one of the main organisations studying and cataloguing attacks. The CERT/CC regularly issues advisories and incident reports. Advisories contain in-depth examinations of attacks, including prevention and the potential impact of the attack. Other organisations, such as the Symantec Corporation, issue similar examinations.

A number of security mailing lists also discuss and examine attacks on a regular basis. Bugtraq[66] is one of the most active and new attacks are presented and analysed via email. Bugtraq is often utilised by other organisations, such as the CERT/CC, as a starting point for attack examination.

For more previous work in examinations of attacks, see the next chapter. The next chapter examines a wide range of attacks and throughout the chapter a number of detailed attack examinations are referred to. As mentioned in the introduction to this chapter, previous work in the taxonomy field can be found in Section 4.2.

2.4 Motivation for Research

There are several motivations for examining computer and network attacks, and proposing a taxonomy for them. As mentioned previously, over the past few years, attacks have increased and become more sophisticated and so pose a significant threat to computer and network users. It is important that attacks are examined closely to help combat them. Also, if a taxonomy is to be proposed, there must be an understanding of the attacks that will be classified.

A taxonomy of computer and network attacks is useful for a number of reasons. While computer and network attacks have become a common occurrence, the language used to describe them is often inconsistent. For example, one information body may label an attack a worm, while another may consider it a virus. Therefore, there needs to be a common language and classification for discussing attacks. A consistent taxonomy should be able to provide this.

A taxonomy will also allow for the applying of previous knowledge to new attacks. If a new attack is identified, and classified appropriately, it should be possible to look at other attacks in the same category to get ideas on how to deal with the new attack.

There are several bodies that will benefit from a taxonomy. Information bodies, Computer Emergency Response Teams (CERTs) and advisory bodies will be able to communicate between themselves more efficiently using a common classification. When a new attack is discovered, if all interested bodies have a common classification, much confusion is avoided.

Chapter 3

Network and Computer Attack Methodologies

The following sections examine some of the types of network and computer attacks. Traditional attacks such as viruses and worms are covered as well as the more recent blended attacks. The categorisation of the attacks in this section is based on the first dimension of the proposed taxonomy, suggested in Chapter 4. It should be noted that many of the attacks described in the following sections are blended and in the proposed taxonomy will have classifications in more than just the first dimension.

Before examining the different types of attacks, the general attack process will be briefly explained. The different network and computer attacks will then be examined, followed by a look at blended attacks.

3.1 The Attack Process

There are several distinct stages that make up an attack on a computer or network, from the initial motivation of the attacker, to the final execution of the attack. In general there are four main stages:

1. Attacker Motivation and Objectives
2. Information Gathering/Target Selection
3. Attack Selection
4. Attack Execution

Howard has a detailed taxonomy built on attack processes, similar to the above stages. His taxonomy is discussed in Section 4.2.3.

While the focus of this research is on the attack, it is important to briefly explain the attack method. An attacker may have many different reasons for launching an attack. Some attackers may simply want to test their skills, others may want to prove a point. Motivation will have some impact on what attacks are chosen and how they are executed.

Before launching the attack, the attacker must select a target and gather information. These two activities take place either concurrently or consecutively, depending on what the attacker wishes to achieve. Information gathering involves extracting useful information from the target network or host, while target selection is the choosing of a promising target. During these stages, the attacker will usually use tools such as packet sniffers and port scanners to gather information on potential targets.

Once the attacker has a target and some information on the potential weaknesses of the target, they can select an attack that is appropriate. The final stage is the execution of the attack, in which the attacker proceeds to launch the attack against the target.

3.2 Viruses

Viruses are self-replicating programs that infect and propagate through files. Usually they will attach themselves to a file, which will cause them to be run when the file is opened. There are several main types of viruses as identified in [68, 64], which are examined below.

3.2.1 File Infectors

File infector viruses infect files on the victim's computer by inserting themselves into a file. Usually the file is an executable file, such as a .EXE or .COM in Windows. When the infected file is run, the virus executes as well.

The Infector Virus is an example of a file infector virus obtained from [65]. The Infector Virus infects .COM files in Windows based systems by attaching itself to the end of the target file. Infection occurs when the infected file is run with the virus selecting one .COM file in the current directory as the target file.

3.2.2 System and Boot Record Infectors

System and boot record infectors were the most common type of virus until the mid 1990s. These types of viruses infect system areas of a computer such as the Master Boot Record (MBR) on hard disks and the DOS boot record on floppy disks. By installing itself into boot records, the virus can run itself every time the computer is booted up. Floppy disks are often infected as users tend to leave floppy disks in the floppy drive. If left in the floppy drive, on reboot the computer may boot from the floppy disk. Thus, the virus has a chance to execute. These types of viruses were very common in the early days of personal computing. However, with the introduction of more modern operating systems, and virus checks being enabled in the Basic Input Output System (BIOS), few of these viruses are being created today. New means of propagation, such as the Internet, are also much more attractive to virus creators.

3.2.3 Macro Viruses

Macro viruses are simply macros for popular programs, such as Microsoft Word, that are malicious. For example, they may delete information from a document or insert phrases into it. Propagation is usually through the infected files. If a user opens a document that is infected, the virus may install itself so that any subsequent documents are also infected. Some macro viruses propagate via email¹, such as the Melissa virus covered in the next section. Often the macro virus will be attached as an apparently benign file to fool the user into infecting themselves.

The Melissa virus[19, 41] is the best known macro virus. It was released in March 1999, and targeted Microsoft Word 97 and 2000. The virus worked by emailing a victim with an email that appeared to come from an acquaintance. The email contained a Microsoft Word document as an attachment, that if opened, would infect Microsoft Word and if the victim used the Microsoft Outlook 97 or 98 email client, the virus would be forwarded to the first 50 contacts in the victim's address book.

Melissa caused a significant amount of damage, as the email sent by the virus flooded email servers. ICISA estimated that Melissa could have caused damage as high as USD \$385 million[53].

The classification of Melissa is interesting. Some consider it a virus, others consider it a worm. Under the proposed taxonomy in Chapter 4, Melissa is considered to be a mass-mailing worm with a viral payload. However, it is included here as the viral payload is a good example of a macro virus. For more information on mass-mailing worms see Section 3.3.1.

3.2.4 Virus Properties

Viruses often have additional properties, beyond being an infector or macro virus. A virus may also be multi-partite, stealth, encrypted or polymorphic.

Multi-partite viruses are hybrid viruses that infect both files and system and/or boot-records. This means multi-partite viruses have the potential to be more damaging, and resistant.

¹ Which makes them type of blended attack.

A stealth virus is one that attempts to hide its presence. This may involve attaching itself to files that are not usually seen by the user.

Viruses can use encryption to hide their payload. A virus using encryption will know how to decrypt itself to run. As the bulk of the virus is encrypted, it is harder to detect and analyse.

Some viruses have the ability to change themselves as either time goes by, or when they replicate themselves. Such viruses are called polymorphic viruses. Polymorphic viruses can usually avoid being eradicated longer than other types of viruses as their signature changes.

3.3 Worms

A worm is a self-replicating program that propagates over a network in some way. Unlike viruses, worms do not require an infected file to propagate. There are two main types of worms: mass-mailing worms and network-aware worms. Each of these is covered in more detail below.

3.3.1 Mass-Mailing Worms

Mass-mailing worms are an interesting category as many attacks in this category could quite easily be classified as a worm, virus or both. For the purpose of this research and the taxonomy, a mass-mailing worm is a worm that spreads through email. Once the email has reached its target it may have a payload in the form of a virus or trojan.

Email, although it may become a file on its journey, is more abstract than a file. Therefore, while some attacks may use email attachments to send viruses, the attack vector² is still email. A case could be made that a mass-mailing virus category would be more appropriate, but the proposed taxonomy attempts to use the attack vector as the first means of classification. Therefore, an attack such as Melissa should be classified first as a mass-mailing worm. For more details on classification see Chapter 4.

3.3.2 Network-Aware Worms

Network-aware worms are a major problem for the Internet. Worms such as SQL Slammer[27] have shown that the Internet can be degraded by a well written worm.

Network-aware worms generally follow a four stage propagation model[14]. Although this is a generalisation, most network-aware worms will fit into this model. Figure 3.1 shows the four stages of network-aware worm propagation from the point of view of a host that is being infected.

The first step is target selection. The compromised host³ targets a host. The compromised host then attempts to gain access to the target host by exploitation. For example, the SQL Slammer worm exploited a known vulnerability in Microsoft SQL Server 2000 and Microsoft Desktop Engine. Once the worm has access to the target host, it can infect it. Infection may include loading trojans onto the target host, creating back doors or modifying files. Once infection is complete, the target host is now compromised and can be used by the worm to continue propagation.

²The attack vector is the way in which an attack reaches its target.

³Or the attacker's computer if the attacker is releasing the worm.

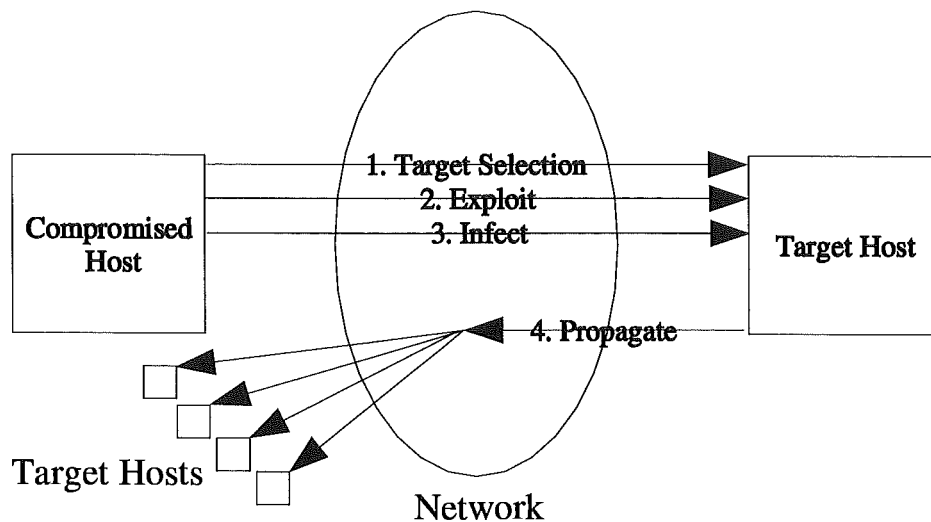


Figure 3.1: The Four Stages of Network-Aware Worm Propagation.

3.4 Trojans

Trojans get their name from the *The Iliad* by Homer, which describes the battle for Troy. Homer writes about how the Greeks created a giant horse, filled it with soldiers, and left it outside Troy. The Trojans, thinking it was a gift of surrender, wheeled the horse inside Troy. At night, the Greek soldiers came out of the horse and opened the gates for the rest of the Greek army. Troy was quickly defeated.

Today's trojans work in a very similar way. They will appear to be benign programs to the user, but will actually have some malicious purpose. Trojans usually carry some payload such as remote access methods, viruses and data destruction.

Back Orifice 2000[60] is a remote administration tool for Windows created by the Cult of the Dead Cow that can be used as a trojan. Back Orifice 2000 (BO2K) is intended to be used by network administrators to manage computers on their network remotely. However, it can be used maliciously. BO2K has the ability to install itself silently so that the user of the computer is unaware that BO2K has been installed. This allows attackers to install BO2K on a target computer without raising suspicion. Once installed BO2K provides a back door for the malicious attacker and gives them the following abilities:

- Session logging.
- Keystroke logging.
- File transfer.
- Program installation.
- Remote rebooting.
- Registry editing.
- Process management.

3.4.1 Logic Bombs

Logic bombs are a special form of trojans that only release their payload once a certain condition is met. For example, a logic bomb might release its payload at a certain time. If the condition is not met, the logic bomb behaves as the program it is attempting to simulate.

3.5 Buffer Overflows

Buffer overflows are probably the most widely used means of attacking a computer or network. They are rarely launched on their own, and are usually part of a blended attack. Buffer overflows are used to exploit flawed programming, in which buffers are allowed to be overfilled. If a buffer is filled beyond its capacity, the data filling it can then overflow into the adjacent memory, and then can either corrupt data or be used to change the execution of the program. There are two main types of buffer overflows described below. More details can be found in [32, 39].

3.5.1 Stack Buffer Overflow

A stack is an area of memory that a process uses to store data such as local variables, method parameters and return addresses. Often buffers are declared at the start of a program and so are stored in the stack. Each process has its own stack, and its own heap (as explained in the next section). Stack overflows are the most common form of buffer overflows.

Overflowing a stack buffer was one of the first types of buffer overflows and is one that is commonly used to gain control of a process. In this type of buffer overflow, a buffer is declared with a certain size. If the process controlling the buffer does not make adequate checks, an attacker can attempt to put in data that is larger than the size of the buffer. This means once the buffer is full, the remaining data being put into it overflows the buffer and overwrites the adjacent memory. An attacker may place malicious code in the buffer. Part of the adjacent memory will often contain the pointer to the next line of code to execute. Thus, the buffer overflow can overwrite the pointer to point to the beginning of the buffer, and hence the beginning of the malicious code. Thus, the stack buffer overflow can give control of a process to an attacker.

3.5.2 Heap Overflows

Heap overflows are similar to stack overflows but are generally more difficult to create. The heap is similar to the stack, but stores dynamically allocated data. The difference between stack allocated data and heap allocated data is shown below:

```
#include <stdlib.h>

int main(){
    char stack_buffer[256];
    char *heap_buffer = (char *) malloc(256 * sizeof(char));
    return 0;
}
```

The heap does not usually contain return addresses like the stack, so it is harder to gain control over a process than if the stack is used. However, the heap contains pointers to data and to functions. A successful buffer overflow will allow the attacker to manipulate the process's execution. An example would be to overflow a string buffer containing a filename, so that the filename is now an important system file. The attacker could then use the process to overwrite the system file (if the process has the correct privileges).

3.6 Denial of Service Attacks

Denial of Service (DoS) attacks [18, 26], sometimes known as nuke attacks, are designed to deny legitimate users of a system from accessing or using the system in a satisfactory manner. DoS attacks usually disrupt

the service of a network or a computer, so that it is either impossible to use, or its performance is seriously degraded. There are three main types of DoS attacks: host based, network based and distributed.

3.6.1 Host Based

Host based DoS attacks aim at attacking computers. Either a vulnerability in the operating system, application software or in the configuration of the host are targeted.

Resource Hog

Some host based DoS are designed to use up (hog) resources on a computer. Resources such as CPU time and memory use are the most common targets. For example, a trivial resource hog is the fork bomb. A fork bomb simply spawns child processes continually, thus over time, more and more resources are taken up by the bomb and its children. A Unix based fork bomb⁴, written in C, is shown below:

```
#include <stdlib.h>

int main(){
    while(1){
        fork();
    }

    return 0;
}
```

Fork bombs, while very effective, are usually easily detected, either through the marked increase in processes, or through logging. They can also be easily prevented by configuring the operating system correctly. Another type of resource hogs access memory in certain patterns, so that thrashing⁵ occurs.

CPU hogs such as Snork[71], exploit vulnerabilities in the operating system. The Snork attack consumes 100% of the target's CPU time. Snork also has a network based DoS component that allows Snork to reduce network bandwidth for legitimate users by continuously bouncing packets between hosts on the network.

Crashers

Crashers are a form of host based DoS that are simply designed to crash the host system, so that it must be restarted. Crashers usually target a vulnerability in the host's operating system. Many crashers work by exploiting the implementation of network protocols by various operating systems. Some operating systems cannot handle certain packets, and if received cause the operating system to hang or crash. Some examples of crashers include Land and Teardrop[17], and the Ping o' Death[49].

3.6.2 Network Based

Network based DoS attacks target network resources in an attempt to disrupt legitimate use. Network based DoS usually flood the network and the target with packets. To succeed in flooding, more packets than the target can handle must be sent, or if the attacker is attacking the network, enough packets must be flooded so that the bandwidth left for legitimate users is severely reduced. Three main methods of flooding have been identified in [26]:

- *TCP Floods*: TCP packets are streamed to the target.
- *ICMP Echo Request/Reply*⁶: ICMP packets are streamed to the target.

⁴When run on a Gentoo Linux 1.4 box, the fork bomb caused an almost instantaneous lock up.

⁵Where more memory pages are accessed than can fit in the physical memory. This results in writing and reading memory pages to and from the hard disk repeatedly, which slows the system significantly down.

⁶Essentially "pinging" the target

- *UDP Floods*: UDP packets are streamed to the target.

In addition to a high volume of packets, often packets have certain flags set to make them more difficult to process. If the target is the network, the broadcast address⁷ of the network is often targeted. One simple way of reducing network bandwidth is through a ping flood. Ping floods can be created by sending ICMP request packets of a large size to a large number of addresses (perhaps through the broadcast address) at a fast rate. On most modern operating systems, root access is required to run the *ping* utility in that way.

3.6.3 Distributed

The last type of DoS attack is perhaps the most interesting. Distributed DoS (DDoS) attacks are a recent development in computer and network attack methodologies. The DDoS attack methodology was first seen in 1999 with the introduction of attack tools such as The DoS Project's Trinoo[36, 21], The Tribe Flood Network[1, 21] and Stacheldraht⁸[37]. Between February 7 and 11, 2000, DDoS attacks were put into the spotlight when DDoS attacks were launched at a number of high-profile web-sites, including Ebay.com, Amazon.com, Yahoo.com and CNN.com. The DDoS attacks were effective enough to disrupt the web-sites' operation for several hours.

DDoS attacks work by using a large number of attack hosts to direct a simultaneous attack on a target or targets. A number of master nodes⁹ are used to control a larger number of daemon nodes¹⁰ which launch the attack on the target. Figure 3.2 shows the process of a DDoS attack. Firstly, the attacker commands the master nodes to launch the attack. The master nodes then order all daemon nodes under them to launch the attack. Finally the daemon nodes attack the target simultaneously, causing a denial of service. With enough daemon nodes, even a simple web page request will stop the target from serving legitimate user requests.

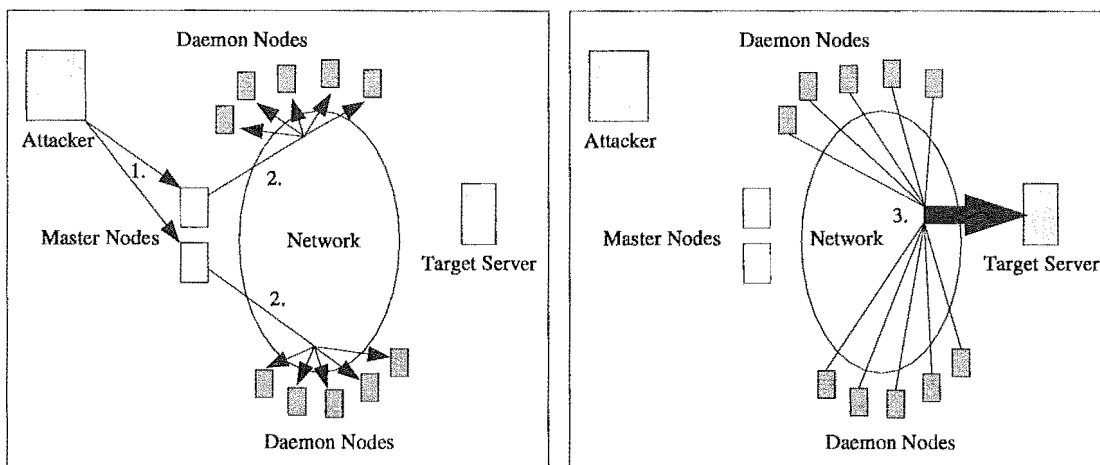


Figure 3.2: The Topology of a DDoS.

3.7 Network-Based Attacks

This section describes several kinds of attacks that operate on networks and the protocols that run the networks.

⁷Any packets sent to the broadcast address get sent to all hosts on a network.

⁸“Barbed-wire” in German.

⁹Hosts directly controlled by the attacker.

¹⁰Subverted hosts that obey the master nodes.

3.7.1 Spoofing

Network spoofing is the process in which an attacker passes themselves off as someone else. There are several ways of spoofing in the standard TCP/IP network protocol stack, including: MAC address spoofing at the data-link layer and IP spoofing at the network layer. By spoofing who they are, an attacker can pretend to be a legitimate user or can manipulate existing communications from the victim host.

MAC Address Spoofing

Medium Access Control (MAC) address spoofing is where the hardware address, that is, the MAC address, is changed so that either the attacker's computer is no longer identifiable as theirs, or the MAC address is the same as a victim's MAC address. This can be used by the attacker to pretend to be someone other than themselves and potentially take over the victim's communications with other computers on the network

In Linux for example, the procedure is simply:

```
bash$ ifconfig eth0 down
bash$ ifconfig eth0 hw ether 00:00:00:00:00:00
bash$ ifconfig eth0 up
```

Where 00:00:00:00:00:00 is the new MAC address. In Windows, the procedure is more complicated and involves modifying the registry¹¹.

MAC address spoofing is only useful to an attacker if their target is on the same subnet as they are. MAC operates at the data-link layer, and so is only used locally. To spoof beyond the local subnet, an attacker must spoof at a higher layer, for example the network layer.

IP Spoofing

Internet Protocol (IP) spoofing is similar to MAC address spoofing described above. However, the attacker's IP address is now spoofed. IP address ranges are often used to determine whether or not a host has access to certain services, so through IP spoofing unauthorised access may be obtained.

IP spoofing is often used to inject commands or data into a existing stream of data between the host and other hosts. To completely take over the data stream, the attacker must change the routing tables so that the packets are routed to the spoofed host¹². More information on IP spoofing can be found in [35].

3.7.2 Session Hijacking

Session hijacking is the process by which an attacker takes over a session taking place between two victim hosts. The attack essentially cuts in and takes over the place of one of the hosts. Session hijacking usually takes place at the TCP layer, and is used to take over sessions of applications such as Telnet and FTP. TCP session hijacking involves use of IP spoofing, as mentioned above, and TCP sequence number guessing.

To carry out a successful TCP session hijacking, the attacker will attempt to predict the TCP sequence number that the session being hijacked is up to. Once the sequence number has been identified, the attacker can spoof their IP address to match the host they are cutting out and send a TCP packet with the correct sequence number. The other host will accept the TCP packet, as the sequence number is correct, and will start sending packets to the attacker. The cut out host will be ignored by the other host as it will no longer have the correct sequence number.

Sequence number prediction is most easily done if the attacker has access to the IP packets passing between the two victim hosts. The attacker simply needs to capture packets and analyse them to determine the sequence number. If the attacker does not have access to the IP packets, then the attacker must guess the sequence number. Sequence numbers are generated in three ways[64]:

1. *64K rule*: The initial sequence counter is incremented with a constant value every second, usually 128 000.

¹¹ Which if done incorrectly could damage the Windows installation.

¹² The spoofed host is the host which has its IP address spoofed to the victim host's address.

2. *Time related generation*: The counter is increased at regular intervals by a number of time-units¹³.

3. *Pseudo-random generation*: The counter is increased by a pseudo-random number.

Prediction is easy when the first method is used. The second is significantly harder, while the third is so hard that most attackers would not bother trying to predict the sequence¹⁴.

Once a session has been hijacked, the attacker is able to do a wide variety of malicious activities. For example, if a Telnet session has been hijacked, the attacker may be able to access the victim's account. Session hijacking is described in more detail in [35, 64].

3.7.3 Wireless Network Attacks

Wireless networks, especially those based on the IEEE 802.11x standards are growing in popularity. However, there are a number of inherent weaknesses in wireless networks that are not an issue in traditional wired networks. Most wireless networks are not configured securely and usually only require MAC address spoofing to gain full access.

WEP Cracking

Wired Equivalent Protocol (WEP) is a standard used by 802.11x networks to encrypt the data transmitted over a wireless network and is widely used. However, the current version of WEP has a flaw that makes it vulnerable. WEP uses a stream cipher¹⁵ to provide encryption and this exposes it to several vulnerabilities (these are examined in detail in [40]). WEP uses a 24-bit initialisation vector (IV) and so, given enough time, the IV is reused for encrypting messages. This reuse can be used to gather information about the encrypted messages. Over time a decryption dictionary can be built to allow the attacker to decrypt the traffic on the network. Of course, to gain enough information to crack the WEP encryption requires time and effort. Fortunately for the war driver, tools exist[62] to automate this procedure.

An in-depth examination of the problems with WEP can be found in [13]. WEP version 2 (WEP2) is being proposed to attempt to solve some of the problems with WEP. The interesting thing to note is that many of the problems with WEP are also a problem with what will be WEP2. WEP2 uses a 104-bit IV, which is still vulnerable to some of the attacks mentioned in [40, 13].

3.7.4 Web Application Attacks

Web application attacks are network attacks that are aimed against web applications. Essentially the application layer of the TCP/IP protocol stack is attacked. Web applications are run through a web browser, but are more than a simple web site. They are usually connected to a database, or at the least have some programs or scripts controlling the web site. An example of a common web application is Internet banking.

Web application attacks are different to attacks that target normal applications, as web applications build upon and use network protocols extensively. Described below are a number of ways in which web applications can be attacked. One form of web application attack is buffer overflows, which are discussed in Section 3.5.

Cross Site Scripting

Cross Site Scripting involves embedding a script within a web application. Usually it occurs on pages that allow for input, such as a guest book or a web forum. The attacker posts a message that contains an embedded script that serves some malicious purpose. For example, the script may prompt other users browsing that page for a user name and password. Other threats include session and account hijacking, cookie theft, and cookie poisoning. More details can be found in The Cross Site Scripting FAQ[3].

¹³Time-units vary and are dependent on how they are measured, how high the CPU load is and so on.

¹⁴Unless they had in-depth knowledge of how the pseudo-random number generator worked and knew the seed value.

¹⁵RC4.

Parameter Tampering

Parameter tampering is a simple web application attack in which the attacker identifies parameters used to drive a web application and modifies a URL header to manipulate the parameters. On a poorly designed site, parameter tampering could be used to maliciously modify stored data. To prevent a parameter tampering attack, parameters should be checked carefully by the web application before processing them.

Cookie Poisoning

Today cookie poisoning is not a large threat, as cookies are usually encrypted. However, it still remains a common form of attack. Cookie poisoning involves modifying a cookie so that the web application is deceived into giving away sensitive data. It is usually used to steal the identity of a user, so that the web application treats the attacker as the victim. Thus, the attacker can access the web application as the victim, and can then gain, damage or delete confidential information.

Database Attacks

Database attacks are web application attacks aimed at accessing the underlying database that drives the web application. The most common form of this type of attack is SQL injection. SQL injection involves submitting a request to the web application with SQL commands appended in a way that the web application passes them on to the database to be processed. For example, suppose the script running the website used the following query (written in PHP):

```
$result = mysql_query  
("SELECT * FROM atable WHERE login='$user' and password='$password'");
```

If the attacker enters a valid user name in the user name field and in the password field enters:

```
password' or 'x'=x
```

Then the query becomes:

```
SELECT * FROM some_table WHERE login='username  
and password='password' or 'x'=x
```

Thus, the password has effectively been made useless, and the attacker can log on to the database as any legitimate user without having to know their passwords. Two more detailed looks at SQL injection can be found in [5, 54].

Hidden Field Manipulation

Hidden field manipulation is a very simple way of attacking a web application. The attacker downloads an HTML page and modifies hidden fields contained in the page. The attacker then reposts the page to the server. Hidden fields may contain important information such as session IDs and user data. Some hidden fields may even contain information such as prices for products being sold through the web applications, so it is possible for an attacker to change prices so that they can buy or sell products at a price that benefits the attacker.

3.8 Physical Attacks

Physical attacks are a form of computer and network attack that are often overlooked in the literature. This may be due to physical attacks being seen as less of a threat. However, physical attacks are often more deadly than other forms of attack and with the rise of interest in Information Warfare will attract more attention. Some forms of physical attacks are very basic, such as cutting a network cable. Attacks such as these will be examined briefly, while the more advanced attacks involving energy weapons and the Van Eck effect will be examined in more detail below.

3.8.1 Basic Attacks

Basic physical attacks on computers and networks can be done by almost anyone. They simply involve using low technological means to cause damage or disruption to a computer or network. There are many different ways an attack could be carried out in this way, for example: cutting a network cable; damaging a computer by hitting it; or using explosives to destroy or disrupt a computer or network.

Because of the nature of these attacks, they are very simple to carry out. However, attacks such as these are not at all subtle, and if someone carried out such an attack it would be hard for them to remain anonymous.

3.8.2 Energy Weapon Attacks

There are currently three main types of energy weapon attacks that can be used to attack computers and networks: high and low energy radio frequency (HERF and LERF) attacks and electro-magnetic pulse (EMP) attacks. While these attacks are more general attacks in that they target the electronics, they are devastating when used against computers and network devices.

HERF weapons focus high energy radio frequency (RF) on a narrow frequency spectrum. HERF can be used quite accurately due to the narrow frequency spectrum. The damage caused by HERF weapons is due to the concentration of energy on electronic components. LERF weapons on the other hand, use a wide frequency spectrum, but with low energy RF. LERF is effective due to the wide frequency range as it is likely that the frequencies will match the resonance frequencies of the target's electronic components.

The Electromagnetic Pulse (EMP) effect was first discovered[42] when the United States was testing high altitude air burst nuclear weapons. The nuclear blast created a very powerful, but short, electromagnetic pulse. When electronic components are exposed to such a pulse, the pulse may create a short transient voltage. The voltage produced can be enough to render the electronic components useless. Nuclear explosions are not the only way to produce an EMP as explained in [51]. EMP bombs can be produced to achieve similar results to a nuclear explosion's EMP.

3.8.3 Van Eck Attacks

The Van Eck effect¹⁶ was popularised by Wim Van Eck in a paper published in 1985[74]. Before the paper was published, it was thought that reconstructing electromagnetic radiation was very difficult and would require expensive equipment and highly trained professionals. Van Eck showed that it was possible to use a television equipped with an extended antenna and two oscillators to reconstruct the signal from a computer monitor. This showed that it was possible for anyone with some electronics knowledge to build such a device and use it to obtain data from a wide range of electronics. By using the Van Eck effect, an attacker can gain sensitive information from the target computer. However, the attacker can gain much more as a recent paper[57] showed. By using optical emanations, the attacker can potentially gain access to data flowing through network equipment.

3.9 Password Attacks

An attacker wishing to gain control of a computer, or a user's account, will often use a password attack[50] to gain the needed password. Many tools[61] exist to help the attacker uncover passwords. There are three ways in which passwords are attacked: by guessing a subset of all possible passwords; by trying all possible passwords; or by exploiting the implementation of the password protection.

3.9.1 Password Guessing/Dictionary Attack

Password guessing is the most simplest of password attacks. It simply involves the attacker attempting to guess the password. This method succeeds more often than would be expected, as many users are

¹⁶This is often referred to as a TEMPEST attack.

predictable in their password choice. Passwords such as names of family members or pets are common. Often the attacker will use a form of social engineering to gain clues as to what the password is.

A dictionary attack is similar, but is a more automated attack. The attacker uses a dictionary of words containing possible passwords and uses a tool to see if any are the required password. Passwords that are English words such as “elephant”, will be very quickly discovered with this form of attack.

3.9.2 Brute Force

Brute force attacks work by calculating every possible combination that could make up a password and testing it to see if it is the correct password. Brute force attacks on passwords are guaranteed to succeed. The only question is how long the brute force attack will take to find the correct password. As the password’s length increases, the amount of time, on average, to find the correct password increases exponentially. This means short passwords can usually be discovered quite quickly, but longer passwords may take decades.

3.9.3 Exploiting the Implementation

Exploiting the implementation involves examining the programs that provide the password protection and finding flaws. If the flaw is significant enough it is possible to circumvent the password protection, or to reveal the password. For example, Microsoft Word 6.0 password protected files can be decrypted[59] quickly by using a flaw in the encryption mechanism.

3.10 Information Gathering Attacks

As mentioned in Section 3.1, the attack process usually involves information gathering. Information gathering is the process by which the attacker gains valuable information about potential targets, or gains unauthorised access to some data without launching an attack. Information gathering is passive in the sense that no attacks are explicitly launched. Instead networks and computers are sniffed, scanned and probed for information.

3.10.1 Sniffing

Packet sniffers are a simple but invaluable tool for anyone wishing to gather information about a network or computer. For the attacker, packet sniffers provide a way to glean information about the host or person they wish to attack, and even gain access to unauthorised information.

Traditional packet sniffers work by putting the attacker’s Ethernet card into promiscuous mode. An Ethernet card in promiscuous mode accepts all traffic from the network, even when a packet is not addressed to it. This means the attacker can gain access to any packet that is traversing on the network they are on. By gathering enough of the right packets the attacker can gain information such as login names and passwords¹⁷.

Other information can also be gathered, such as MAC and IP addresses and what services and operating systems are being run on specific hosts. This form of attack is very passive. The attacker is not sending any packets out, they are only listening to packets on the network.

3.10.2 Mapping

Mapping is used to gather information about hosts on a network. Information such as what hosts are on-line, what services are running and what operating system a host is using, can all be gathered via mapping. Thus potential targets and the layout of the network, are identified

Host detection is achieved through a variety of methods. Simple ICMP queries can be used to determine if a host is on-line. TCP SYN messages can be used to determine whether or not a port on a host is open and thus, whether or not the host is on-line. Host detection techniques are discussed in detail in [6].

¹⁷Telnet is notoriously insecure in this regard. Telnet sends passwords unencrypted, so if the correct packets are captured, the password can be extracted.

After detecting if a host is on-line, mapping tools can be used to determine what operating system and what services are running on the host. There are a wide range of techniques that can be used. Simply examining the service banners¹⁸ may reveal the operating system. More advanced techniques include analysing the network protocol stack used by the operating system.

Running services are usually identified by attempting to connect to a host's ports. Port scanners are programs that an attacker can use to automate this process. Basic port scanners work by connecting to every TCP port on a host and reporting back which ports were open. More sophisticated port scanners, such as Nmap[48], use additional techniques to avoid detection and to gain more information.

Mapping identifies potential targets, such as a version 6.0 IIS web server, but specific vulnerabilities that could be exploited are not identified. Either the attacker has to choose an attack using the information gathered, or more information needs to be gathered through security scanning, discussed below.

3.10.3 Security Scanning

Security scanning is similar to mapping, but is more active and more information is gathered. Security scanning involves testing a host for known vulnerabilities or weaknesses that could be exploited by the attacker. For example, a security scanning tool may be able to tell the attacker that port 80 of the target is running an HTTP server, with a specific vulnerability. Security scanning is more easily detected than mapping, as attack patterns testing the vulnerabilities can usually be detected by intrusion detection systems.

3.11 Blended Attacks

While blended attacks are not a new development, they have recently become popular with attacks such as Code Red and Nimda. Blended attacks are attacks that contain multiple threats, for example multiple means of propagation or multiple attack payloads. Many of the attacks mentioned previously in this chapter can be considered as blended.

The first instance of a blended attack occurred in 1988 with the first Internet worm: the Morris Worm. The Morris Worm attacked and propagated through multiple vulnerabilities in Unix based systems. Newer attacks such as Code Red and Nimda work in a similar way by exploiting multiple vulnerabilities and by launching multiple attacks. For in-depth analysis of the Morris Worm see [38, 67].

Code Red[23] is the most famous blended attack. It was the first of the new wave of blended attacks and it came as a surprise to the security industry. Code Red was also the first worm to spread through memory rather than through file uploads. Microsoft's Internet Information Services (IIS) web server was Code Red's target. IIS versions from 4.0 to 6.0b all contained a buffer overflow vulnerability[22] in the Indexing Service DLL of IIS. Code Red spread by using a buffer overflow to compromise susceptible hosts and once a host was infected, Code Red would do the following, depending on which day of the month it was:

- *Day 1 - 19*: Code Red would try to spread by attempting to connect to vulnerable hosts.
- *Day 20 - 27*: A denial of service attack would be launched against a fixed IP address.
- *Day 28 - end of month*: No activity.

Code Red is a blended attack as it is a worm that utilises a buffer overflow attack and launches a denial of service attack. Three separate attacks are combined together to produce a dangerous blended attack. Code Red is discussed further in [9, 75].

Blended attacks have become one of the leading security threats and will no doubt continue to be a significant problem in the future. While blended attacks have existed for some time, a new wave of highly damaging attacks started with the release of Code Red. The Internet is especially susceptible to blended threats, as was shown by the recent SQL Slammer[27] attack, in which the Internet suffered a significant loss of performance.

¹⁸Services sometimes display the operating system name and version in a welcome banner.

Chapter 4

Toward a Taxonomy

In this chapter the taxonomy is proposed. Before describing the taxonomy, the requirements for the taxonomy and previous work in this field are discussed in Sections 4.1 and 4.2. Finally, Section 4.3 details the process of creating the taxonomy as well as how the taxonomy works.

4.1 Requirements of a Taxonomy

Before examining existing taxonomies and working toward a new one, it is important to define what a good taxonomy consists of. In other words, what are its requirements? For a taxonomy to be useful it has to meet some basic requirements. If, for example, it is not repeatable, then the taxonomy would fail to be useful. Therefore, it is crucial that the taxonomy's requirements are defined. A number of requirements that have been compiled from various sources in [56], provide a good starting point. Below are some of the requirements that are relevant to the proposed taxonomy:

- *Accepted*[4, 44]: The taxonomy should be structured so that it can become generally approved.
- *Comprehensible*[55]: A comprehensible taxonomy will be able to be understood by those who are in the security field, as well as those who only have an interest in it.
- *Completeness*[4] / *Exhaustive*[44, 55]: For a taxonomy to be complete/exhaustive¹, it should account for all possible attacks and provide categories for them. While it is hard to prove a taxonomy is complete or exhaustive, they can be justified through the successful categorisation of actual attacks.
- *Determinism*[52]: The procedure of classifying must be clearly defined.
- *Mutually exclusive*[44, 55]: A mutually exclusive taxonomy will categorise each attack into, at most, one category.
- *Repeatable*[44, 52]: Classifications should be repeatable.
- *Terminology complying with established security terminology* [55] : Existing terminology should be used in the taxonomy so as to avoid confusion and to build on previous knowledge.
- *Terms well defined*[12]: There should be no confusion as to what a term means.
- *Unambiguous*[44, 55]: Each category of the taxonomy must be clearly defined so that there is no ambiguity as to where an attack should be classified.
- *Useful*[44, 55]: A useful taxonomy will be able to be used in the security industry. For example, the taxonomy should be able to be used by incident response teams.

The taxonomy is proposed later on in this chapter. In Chapter 5 the taxonomy is tested against the above requirements.

¹Completeness and exhaustiveness are essentially the same requirement.

4.2 Existing Taxonomies and Previous Work

The field of network and computer security has seen a number of taxonomies aimed at classifying attacks. In the following section some of the more prominent taxonomies will be examined.

4.2.1 Early Security Taxonomies

The two most important early taxonomies in the security field were the Protection Analysis[47] (PA) taxonomy and the Research in Secured Operating Systems[2] (RISOS). While they focus on vulnerabilities rather than attacks, they provide a good background to proposing new taxonomies. Both focused on categorising security flaws and both resulted in similar classification schemes. Each consisted of a number of classes that are roughly equivalent.

As Bishop points out in [10], both taxonomies suffer from ambiguity between the classes. Some vulnerabilities may fall across multiple classes and therefore the taxonomies will not be mutually exclusive. However, the concepts from these early taxonomies are valuable, and have been used in newer taxonomies ([56, 11, 7]). Comparisons of the two taxonomies can be found in [11, 10, 56].

4.2.2 Bishop's Vulnerability Taxonomy

Matt Bishop has made several important contributions to the field of security taxonomies. In [11], Bishop presents a taxonomy of Unix vulnerabilities in which the underlying flaws of vulnerabilities are used to create a classification scheme. Six "axes" are used to classify vulnerabilities:

- *Nature*: The nature of the flaw is described using the Protection Analysis categories.
- *Time of introduction*: When the vulnerability was introduced.
- *Exploitation Domain*: What is gained through the exploitation.
- *Effect Domain*: What can be affected by the vulnerability.
- *Minimum Number*: The minimum number of components necessary to exploit the vulnerability.
- *Source*: The source of identification of the vulnerability.

Bishop's approach is interesting, as instead of a flat or tree-like taxonomy, he uses axes. In the proposed taxonomy a similar structure is used as described in Section 4.3.

Bishop also performed a critical analysis of other vulnerability taxonomies in [10]. Previous taxonomies such as PA, RISOS and Aslam's taxonomy[7] are assessed and compared. He also examines the issues surrounding taxonomies and especially what makes a good taxonomy. Bishop suggests that one of the main benefits of a taxonomy is that it should help to work out where to invest resources.

4.2.3 Howard's Taxonomy

In [44], John Howard presents a taxonomy of computer and network attacks. The approach taken is broad and process-based, taking into account factors such as attacker motivation and objectives.

Figure 4.1 shows Howard's taxonomy. The taxonomy consists of five stages: attackers, tools, access, results and objectives. The attackers consist of a range of types of people who may launch an attack. These range from hackers to terrorists. Tools are the means that the attackers use to gain access. Access is gained through either an implementation, design or configuration vulnerability. Once access is gained, the results may be achieved such as corruption or disclosure of information. From this process the attacker achieves their objectives which may vary from inflicting damage, to gaining status.

In regards to the computer and network attack taxonomy that is suggested in Section 4.3, the tools stage of Howard's taxonomy is roughly analogous. The proposed taxonomy is focused solely on the attacks, rather than the attack process.

Howard attempts to focus attention on a process driven taxonomy, rather than a classification scheme such as in the animal kingdom. This means the whole attack process is considered, which is certainly

valuable. However, as Lough points out in [56], Howard fails to meet one of his taxonomy requirements: mutual exclusion. Some of the categories shown in Figure 4.1 may overlap. For example the attacker's category contains classes that may not be mutually exclusive. As Lough points out:

“Depending on one’s point of view, a *terrorist’s* actions could be indistinguishable from those of a *vandal*. A *spy* could be a *professional criminal*.”

Howard’s approach is still useful in gaining insight to the process of attacks. However, for information bodies such as CERT, such a taxonomy may not be practical. Information bodies are more concerned with the attack itself, than with the motivations and objectives behind it.

Some of Howard’s ideas have been applied in the proposed taxonomy, notably in the third (Section 4.4.3) and fourth (Section 4.4.4) dimensions of the proposed taxonomy.

Howard extends his work further in [45] by refining some of the stages. However, the problems mentioned above still exist with the refined taxonomy.

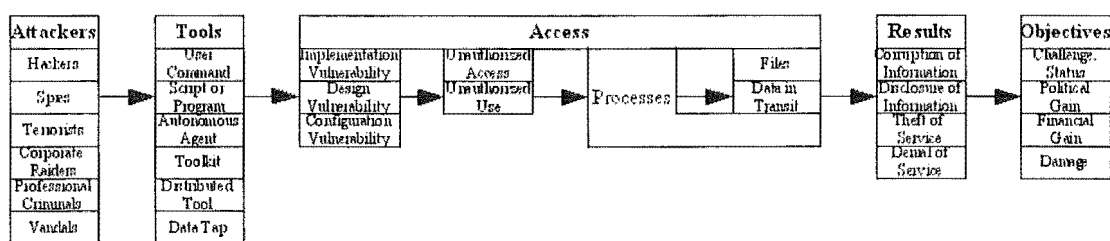


Figure 4.1: Howard’s Process Based Taxonomy.

4.2.4 Lough’s Taxonomy

In 2001, Daniel Lough proposed another taxonomy named VERDICT[56]. VERDICT stands for Validation Exposure Randomness Deallocation Improper Conditions Taxonomy and is based on characteristics of attacks. Instead of a tree-like taxonomy, Lough proposed using four characteristics of attacks:

- *Improper Validation:* Insufficient or incorrect validation results in unauthorised access to information or a system.
- *Improper Exposure:* A system or information is improperly exposed to attack.
- *Improper Randomness:* Insufficient randomness results in exposure to attack.
- *Improper Deallocation:* Information is not properly deleted after use and thus can be vulnerable to attack.

Lough proposes that any attack can be classified using these four characteristics. By basing the taxonomy on characteristics, the taxonomy can easily and tidily classify blended attacks. Lough’s approach is similar to both Bishop’s axes and to the proposed taxonomy’s dimensions.

Lough’s taxonomy is interesting and has influenced the proposed taxonomy. However, there are a few shortcomings to Lough’s taxonomy. While it is useful for applying to a new technology (Lough applies it to 802.11 and finds numerous vulnerabilities) to discover new vulnerabilities and to classify existing ones, it may be helpful to have a more specific taxonomy.

In terms of an information body, Lough’s taxonomy may not be useful for the day to day task of identifying and classifying new attacks, and issuing advisories. Lough’s taxonomy is general, and does not speak about attacks in terms of worms, viruses, and trojans, which is how attacks are usually described.

In the end, the goals of the taxonomy determine its usefulness. The proposed taxonomy aims to be a practical, specific taxonomy that can be used by information bodies to classify new attacks. Lough’s taxonomy on the other hand, succeeds in providing a taxonomy that is useful for analysis and for the prediction of new attacks.

4.2.5 OASIS Web Application Security Technical Committee

The OASIS Web Application Security Technical Committee[72] (OASIS WAS TC) is a current attempt to provide a classification scheme for web application vulnerabilities. It is a new initiative with the first meeting on 3 July 2003. Currently it is being developed and is in the early stages of being drafted. OASIS WAS TC is leaning toward using attack vectors as the first step of classification, in a similar way to what is suggested in the proposed taxonomy². XML is being used to describe vulnerabilities so that interoperability is enhanced.

It will be interesting to see how the OASIS WAS TC progresses over the next few years. While still in its early stages, it has produced some good ideas and there is active discussion on the committee's mailing lists[73].

4.3 The Proposed Taxonomy

While the taxonomies discussed in the previous section are useful, they tend to be general in their approach to classifying attacks. Taxonomies such as Howard's (Section 4.2.3) give a good overview of the attack process, but avoid examining the categories of attacks that face computers and networks each day. For example, classifying attacks such as the Code Red worm would be hard to do using Howard's taxonomy. Therefore, there is a need for a taxonomy that allows for specific kinds of computer and network attacks, such as worms, viruses and buffer overflows. The goal is to provide a pragmatic taxonomy that is useful to those dealing with attacks on a regular basis.

During the taxonomy's development, several model taxonomies were attempted without success. The initial approach was to create a taxonomy analogous to the animal kingdom's taxonomy. The resulting taxonomy would be a tree-like structure with the more general categories at the top, and specific categories at the leaves. However, while such a taxonomy is certainly desirable, in practise it is not possible to do so in an acceptable manner.

The first problem with such a taxonomy is how to deal with blended attacks. To allow for attacks to contain other attacks there are two possible solutions. One is to allow for cross-tree references, that is when one leaf node points to another leaf node somewhere else in the taxonomy. This approach leads to a messy tree and would be hard to use in classifying. The second is to have recursive trees, so that each leaf on the base tree may have another tree (or more) under it. This again leads to a messy structure and would be of limited use.

The second problem is that attacks, unlike animals, often do not have many common traits. This makes the creation of broad categories hard. While worms and viruses can be related³, there is little in common between them and a buffer-overflow. This means that the taxonomy tree would have to branch out immediately into a number of categories that are unrelated. The benefits of the tree-like structure are therefore lost. With these two problems, the tree-like taxonomy was discarded.

Another way taxonomies are sometimes created, is through lists. A list based taxonomy contains a flat-list of categories. There are two approaches that could have been taken in the proposed taxonomy. Firstly, a flat-list with general categories could be suggested, or secondly, a flat-list with very specific categories could be proposed. The problem with the first case is that general categories are of limited use. In the domain of network and computer attacks, the categories would have to be very general to accommodate the problem of blended attacks. Such a general taxonomy will not be very useful. The second case also suffers from the problem of blended attacks. If very specific categories were chosen, such that any type of blended attack had a category, the list would become almost infinite, with few instances within each category.

The proposed taxonomy takes a different approach than both the tree-like taxonomy or the flat-list taxonomy. However, both these approaches are used by the proposed taxonomy as parts of the complete taxonomy. The next section explains this in detail.

²The idea of attack vectors for the taxonomy, was explored before researching the OASIS WAS TC.

³As both are self-replicating.

4.3.1 Overview

The proposed taxonomy works by using the concept of dimensions. Dimensions are a way of allowing for a classification of an attack to take a more holistic view of the attack. The taxonomy proposes four dimensions for attack classification. Before examining how the taxonomy works, the dimensions used are briefly explained.

The first, or base, dimension is used to categorise the attack into an attack class that is based on the attack vector, or if there is no attack vector, the attack is classified into the closest category.

The attack target is covered in the second dimension. The target can be classified down to very specific targets, such as Sendmail 8.12.10 or can cover a class of targets, such as Unix based systems.

The third dimension covers the vulnerabilities and exploits, if they exist, that the attack uses. The vulnerabilities and exploits do not have a structured classification due to the possible infinite number of vulnerabilities and exploits. Instead the list defined by the Common Vulnerabilities Exposures project[34] is used as a starting point.

The fourth dimension takes into account the possibility for an attack to have a payload or effect beyond itself. In many cases an attack will be clearly a certain kind of attack, but yet it will have a payload or cause an effect that is different. For example, a virus that installs a trojan horse, is still clearly a virus, but has a trojan as a payload.

In each dimension, the classifier must classify attacks as specifically as possible. This means attacks should be classified down to the smallest sub-class in each dimension that makes sense.

The taxonomy allows for the possibility of further dimensions which, although not necessary, may enhance the knowledge of the attack. Some further dimensions are discussed in Section 4.4.5.

An attack must have at least the first dimension, but depending on the attack, or how specific the classifier wishes to be, all, some or none of the other dimensions may be used. The next section explains the details of each dimension and how they work to provide a classification.

4.4 Classification

4.4.1 The First Dimension

Classification in the first dimension consists of two options:

- If the attack uses an attack vector, categorise by the vector.
- Otherwise find the most appropriate category, using the descriptions for each category below.

The attack vector of an attack is the main means in which the attack reaches its target. For example, the Melissa “Virus” uses email as its main form of propagation, and therefore is, in the first dimension, a mass-mailing worm. The virus-like capabilities of Melissa are handled in the other dimensions.

It is very important that attack vectors are identified if possible, as they provide the most accurate description of an attack. For example, an attack that infects computers through a TCP network service and then installs a trojan on the infected computer, should be classified by its attack vector - which is a worm (it spreads through via network services). If it is classified as a trojan instead, then there is no opportunity to describe the worm-like behaviour of the attack, which is essentially the most important feature of the attack.

If an attack vector is not present or is too trivial⁴ then the attack can be categorised by finding the category closest to how the attack works. For example, an attack run locally that gains control of another process by overflowing a buffer, is a buffer overflow.

⁴That is, the vector is outside the categories defined in the first dimension.

Table 4.1: The First Dimension's Categories

Viruses:	File Infectors System/Boot Record Infectors Macro	
Worms:	Mass Mailing Network Aware	
Trojans:	Logic Bombs	
Buffer Overflows:	Stack Heap	
Denial of Service Attacks:	Host Based: Network Based: Distributed	Resource Hogs Crashers TCP Flooding UDP Flooding ICMP Flooding
Network Attacks:	Spoofing Session Hijacking Wireless Attacks: Web Application Attacks:	WEP Cracking Cross Site Scripting Parameter Tampering Cookie Poisoning Database Attacks Hidden Field Manipulation
Physical Attacks:	Basic Energy Weapon: Van Eck	HERF LERF EMP
Password Attacks:	Guessing: Exploiting Implementation	Brute Force Dictionary Attack
Information Gathering Attacks:	Sniffing: Mapping Security Scanning	Packet Sniffing

Chapter 3 gives more detail on each of the categories shown in Table 4.1. However, to help categorise attacks if they do not have an obvious attack vector, the following definitions are given. When categorising, choose the the category that matches best with the definitions below. Once the general class has been chosen, the attack may be further classified by using the sub-classes, if they exist. See Chapter 3 for more information on each of the sub-classes.

- *Virus*: Self-replicating program that propagates through some form of infected files.
- *Worms*: Self-replicating program that propagates without using infected files. Usually worms propagate through network services on computers or through email.

- *Trojans*: A program made to appear benign that serves some malicious purpose.
- *Buffer Overflows*: A process that gains control or crashes another process by overflowing the other process' buffer.
- *Denial of Service Attacks*: An attack which prevents legitimate users from accessing or using a host or network.
- *Network Attacks*: Attacks focused on attacking a network or the users on the network by manipulating network protocols, ranging from the data-link layer to the application layer.
- *Physical Attacks*: Attacks based on damaging physical components of a network or computer.
- *Password Attacks*: Attacks aimed at gaining a password.
- *Information Gathering Attacks*: Attacks in which no physical or digital damage is done and no subversion occurs, but in which important information is gained by the attacker, possibly to be used in a further attack.

The first dimension is summarised in Table 4.1. The categories are reasonably broad. To categorise more specifically, other dimensions need to be used.

The categories that can be used as attack vectors are: viruses, worms and trojans. These categories have the necessary characteristics⁵ to be vectors. While it may not be impossible to use another category as an attack vector, it should be a rare occurrence and would suggest that an incorrect classification has been made.

4.4.2 The Second Dimension

The second dimension covers the target(s) of the attack. As an attack may have multiple targets, there may be multiple entries in this dimension.

It is important to note that targets should be made specific. That is, for an attack on Server A, we are not concerned that Server A was attacked. Rather the operating system of Server A and service that was attacked are important. So for example, if Code Red attacked Server A, the target would not be Server A, but the IIS server that Server A was running.

Table 4.2 shows the categories of the second dimension. Note that Table 4.2 is not complete. There are a wide range of potential targets and each year the list increases. Instead, what is presented is a generalised way of classifying the targets with a few specific examples. The entries in Table 4.2 that contain “...” show where extra categories can be added to the classification. Extra entries should be added in a way that conforms with how the sibling categories have been defined. For example, if adding a category for the DOS operating system, firstly a “DOS Family” entry should be created under Software→Operating System, then the flavours of DOS should be created within the “DOS Family” entry. Finally, within each flavour of DOS entry, specific versions should be created.

Hardware targets can be put into three main sub-classes: computer, network equipment and peripheral devices. Computer targets are computer components, such as CPUs and hard-disks. Network equipment targets are network hardware such as hubs, or network cable. Finally, peripheral devices are devices that are not essential⁶ to a computer, for example monitors.

Software targets have two main classes: operating system and application targets. Operating system targets are targets within the operating system itself, while application targets are targets that are running on top of the operating system.

Finally, a network target is when the network itself or its protocols are targeted. For example, a ping-flood attacks a network rather than hardware or software.

⁵Such as having the ability to carry other attacks.

⁶Essential devices are ones that the computer could not operate without. For example, the CPU and memory are essential.

Table 4.2: The Second Dimension's Categories

Hardware:	Computer:	Hard-disks				
		...				
	Network Equipment:	Hub				
		Cabling				
		...				
	Peripheral Devices:	Monitor				
		Keyboard				
		...				
	Software:	Operating System:	Windows Family:	Windows XP		
				Windows 2003 Server		
			...			
		Unix Family:	Linux:	2.2		
				2.4		
				...		
			FreeBSD:	4.8		
				5.1		
				...		
				...		
		MacOS Family:	MacOS X:	10.1		
				10.2		
				...		
				...		
				...		
	Application:	Server:	Database			
			Email:			
			Web:	IIS:	4.0	
					5.0	
				...		
		User:	Wordprocesor:	MS Word:	2000	
					XP	
					...	
			Email Client	...		
				...		
				...		
Network:	Protocols:	Transport-Layer:	IP			
				...		
		Network-Layer:	TCP			
				...		
				...		

4.4.3 The Third Dimension

The third dimension covers the vulnerabilities and exploits that the attack uses. An attack may exploit multiple vulnerabilities, so there may be more than one entry in the third dimension. Entries in the third dimension are usually a Common Vulnerabilities and Exposures (CVE) entry, but in the case that a CVE entry does not exist, the vulnerability is classified generally as described later on in this section.

The Common Vulnerabilities and Exposures project[34] is designed to produce common definitions of vulnerabilities. The idea for CVE was proposed by Mann and Christey in [58]. The CVE project has become the de facto standard for vulnerabilities and so it is desirable that the proposed taxonomy utilises

this. It should be noted that vulnerabilities are wide and varied and usually apply to specific versions of a piece of software or operating systems. This means a classification scheme would have to include every piece of software in use today.

Below is an example of a CVE entry showing a vulnerability in Microsoft Internet Information Services which the Code Red worm exploited.

Name: CVE-2001-0500
Description: Buffer overflow in ISAPI extension (idq.dll) in Index Server 2.0 and Indexing Service 2000 in IIS 6.0 beta and earlier allows remote attackers to execute arbitrary commands via a long argument to Internet Data Administration (.ida) and Internet Data Query (.idq) files such as default.ida, as commonly exploited by Code Red.
References: <ul style="list-style-type: none">• BUGTRAQ¹: 20010618 All versions of Microsoft Internet Information Services,• Remote buffer overflow (SYSTEM Level Access)• MS²: MS01-033• CERT³: CA-2001-13• BID⁴: 2880• XF⁵: iis-isapi-idq-bo(6705)• CIAC⁶:L-098

Figure 4.2: CVE Entry: CVE-2001-0500

Once the vulnerability or vulnerabilities that an attack exploits are known, the relevant CVE entries can be found. Howard suggests three general types of vulnerabilities in [44]:

- *Vulnerability in implementation*: The design of the system is secure, but the implementation fails to meet the design and thus vulnerabilities are introduced.
- *Vulnerability in design*: The fundamental design of the system is flawed, so that even a perfect implementation will have vulnerabilities.
- *Vulnerability in configuration*: The configuration of the system introduces vulnerabilities. The system itself may be secure but if configured incorrectly, renders itself vulnerable.

If no CVE entry exists, then one of Howard's types of vulnerabilities should be selected, and a description of the vulnerability should be created. As time progresses, CVE entries may be added, in which case classifications may have to be updated to reflect this.

¹BUGTRAQ mailing list (<http://www.securityfocus.com/archive/1>).

²Microsoft Security Bulletin (<http://www.microsoft.com/security/bulletins/current.asp>).

³CERT/CC Advisory (<http://www.cert.org/advisories>).

⁴Security Focus Bugtraq ID database entry (<http://online.securityfocus.com/bid>).

⁵X-Force Vulnerability Database (<http://xforce.iss.net>).

⁶Department of Energy Computer Incident Advisory Center bulletins (<http://ciac.llnl.gov/cgi-bin/index/bulletins>)

4.4.4 The Fourth Dimension

The third dimension deals with attacks having payloads or effects beyond themselves. For example, a worm may have a trojan payload, or it may simply destroy some files. The payload may be another attack itself and so the first dimension can be used to classify the payload if this is the case.

The fourth dimension consists of five categories:

1. First Dimension Attack Payload (see Section 4.4.1)
2. Corruption of Information
3. Disclosure of Information
4. Theft of Service
5. Subversion

Categories 2-4 were previously identified by Howard in [44]. Corruption of information occurs when a payload corrupts or destroys some information. When a payload discloses information that is not intended by the victim to be disclosed, the payload is a disclosure of information payload. Theft of service payloads use a system's services without authorisation, but without impacting the service of legitimate users. Howard has a fourth category, denial of service. However, this possibility is covered in Category 1. Finally, a subversion payload will gain control over part of the target and use it for its own use.

It should be noted that apart from the First Dimension Attack Payload, the categories are general. This is because while general types of payloads can be identified, there are a wide range of implementations of the various payloads. For example, two attacks may corrupt information in that they delete files, but may only differ in which files they delete. In most cases it should be possible to use a first dimension category as the payload.

4.4.5 Other Dimensions

Besides the four dimensions described above, a number of further dimensions could be added to enhance the taxonomy. Several are discussed below and although they are more abstract and are not as essential as the previous dimensions, they are still useful in classifying attacks, especially in regards to how to react to a new attack that falls into a certain category.

For example, the following are dimensions that would be useful for an organisation dealing with attacks:

- *Damage*: A damage dimension would attempt to measure the amount of damage that the attack does. Attacks have different degrees of damage. An attack such as the recent SoBig virus⁷ cause more damage than a simple virus such as the Infector virus (see Section 3.2.1).
- *Cost*: Cleaning up after an attack costs money. In some cases billions of dollars are spent on attack recovery.
- *Propagation*: This category applies more to replicating attacks. The propagation of an attack is the speed at which it reproduces or spreads. For attacks such as worms and viruses, a dimension covering this aspect would be useful.
- *Defence*: The methods in how an attack has been defended against could be made into a further defence dimension.

It should be noted that the new dimensions suggested above are post-attack dimensions. That is, the attack will have to have had time to show its attack potential, so that an accurate assessment of the damage or cost can be made. The four base dimensions however, can be applied relatively soon after the attack has been launched. There is also the possibility for classification refinement, so that as more information is known about an attack, the classification is made more specific.

⁷And it's variants.

Chapter 5

Evaluation of the Proposed Taxonomy

5.1 Introduction

The purpose of this chapter is to briefly evaluate the taxonomy proposed in Chapter 4. Before starting the taxonomy, a list of requirements (see Section 4.1) were compiled from the literature that the taxonomy should ideally meet. These requirements are tested in this chapter. Also, a number of attacks were classified using the proposed taxonomy, to show how the taxonomy is applied practically. Appendix A contains the classifications that were made.

Section 5.2 examines the classification process to see how the proposed taxonomy works in practise and whether or not the requirements are met. In Section 5.3, future work is described, detailing how the taxonomy could be improved and what areas would be interesting to research further.

5.2 Analysis

Before examining the requirements that the taxonomy is supposed to meet, a brief analysis of the classification process is given. In general, it was found the taxonomy worked well and that most attacks were easily (with the appropriate information) classified.

However, there were a number of issues that were identified:

- *Blended Attacks*: While the taxonomy deals with blended attacks well, some blended attacks (especially Nimda) were hard to classify. This was due to the complexity of the attacks as they contained multiple sub-attacks.
- *Targets*: The second (target) dimension overall worked well. However, in some cases it was hard to determine what the target was. For example, a worm like Nimda attacks specific versions of Internet Explorer (IE) but email clients were affected the most¹. However, as described in Section 4.4.2, attacks must made specific, that is, it is the specific versions of IE that are being attacked and not the email clients.
- *Blended Sub-Attacks*: One problem occurred when classifying the Melissa attack. The Melissa attack contains a macro virus payload in a Microsoft Word document. The document is a trojan in the sense that it appears to be benign. The taxonomy was unable to account for both the payload being a virus and a trojan. However, the main feature of the payload is that it is a virus, therefore Melissa was categorised in the fourth dimension as a macro virus.
- *Ranges*: Ranges of classifications, especially in the second (target) dimension could be handled better. Ranges such as DOS versions 2.4 to 4.1 require every DOS version in the range to be added to the classification.

¹As many email clients use IE to view HTML emails.

- *Requirements*: One problem that the taxonomy cannot handle fully is when an attack requires a combination of targets to be successful. For example, an attack may require that a certain operating system run a certain service. If the service and operating system are not in the certain combination, then the attack fails. Thus, there is a relationship between the two targets. This relationship is currently not accounted for, so in the above situation, each target will simply be listed in the second dimension. The same problem exists for the third (vulnerabilities) dimension.

5.2.1 Requirements

In Section 4.1, a number of requirements were defined that the proposed taxonomy should meet. In this section, those requirements are re-examined in regards to the proposed taxonomy to see whether or not the taxonomy meet the requirements.

Accepted

For the proposed taxonomy to be accepted, it should be structured so that the general security community is able to accept it. The proposed taxonomy builds on previous work that is well accepted in the security community, and utilises projects, such as the CVE, that are well respected. While it remains to be seen if the proposed taxonomy is accepted, it is certainly acceptable. Improvements described in Section 5.3, will further make the proposed taxonomy acceptable to the security community.

Comprehensible

A comprehensible taxonomy is one that can be understood by both security experts, and those with a slight interest in the field. Generally the taxonomy is comprehensible: the splitting of attacks into four dimensions separates the attack into understandable components.

However, there are some areas which could be improved. It would be preferable to have names that accurately described each dimension, rather than labelling each dimension with a number. While this is straight-forward for dimensions two to four: Target, Vulnerability and Payload respectively; the base dimension presents more of a problem. The base dimension classifies the attack payload or the most striking feature of the attack, and so naming it is non-trivial.

Completeness/Exhaustive

Completeness a hard requirement to prove. A range of attacks were categorised in the above sections, and although only a few attacks are covered, they do help to show that the taxonomy is complete to a certain extent. However, the nature of the proposed taxonomy is that it can be extended. All current types of attacks are covered, and if new ones are introduced, then the taxonomy can be extended to cover them. The taxonomy is flexible in this sense, so can be adapted if found not to be complete.

Determinism

The procedure by which classification occurs is clearly defined in Chapter 4. However, it was noticed that while classifying attacks, sometimes it was hard to determine what the attack consisted of. For example, when an attack had no obvious attack vector, then sometimes it was difficult to determine which category was the closest. However, the difficulty was not major and after some thought it became clear which was the appropriate category.

Mutually Exclusive

Mutually exclusiveness means that each attack can only be categorised into, at most, one category. The proposed taxonomy, through the defined procedure, does not allow for attacks to be categorised into multiple categories. It does however allow for the refinement of a classification. So an attack may be categorised generally initially, but as more information about the attack becomes evident, the categorisation can be refined. However, at no time should the same attack be classified in two different ways.

Repeatable

Repeatability means that classifications of an attack should be repeatable. This requirement links in strongly with determinism. If the procedure is clearly defined, then the taxonomy should be repeatable. As the procedure is defined carefully, in most instances classifications will be repeatable. In the rare occasions where an attack is classified differently by different people, the procedure in Chapter 3 should be examined to find out which classification is correct.

Terminology Complying with Established Security Terminology

Security terminology in regards to many types of computer attacks (especially viruses and worms) is not well established. Sometimes one attack will be described as a worm, while elsewhere it is described as a virus. The proposed taxonomy kept to commonly used terminology as much as possible, but where there was ambiguity, categories were specifically defined. The definitions will hopefully help in removing some of the ambiguity when certain attacks are described.

The proposed taxonomy uses the Common Vulnerabilities and Exposures (CVE) project as a basis for the third dimension. The CVE project is well established and provides terminology for describing vulnerabilities. Thus, the taxonomy builds on existing terminology.

Terms well defined

In general the taxonomy's categories were found to be well defined. Given the procedure described in Section 4.3, identifying whether an attack had an attack vector or not, was relatively simple. One area that could be improved is the network attacks category. The definition used for this category is quite general, and a tighter definition would benefit the taxonomy.

Unambiguous

Unambiguity means that the taxonomy must have clearly defined classes. There should be no doubt as to which class an attack belongs to. The proposed taxonomy's use of dimensions means that classifications are less ambiguous as different aspects of attacks are covered in each dimension. Therefore, when classifying under the first dimension, the classifier only has to think about whether the attack has a vector and which class it is closest to. Thus, concerns are separated which means it is less likely that there will be ambiguity.

Useful

Usefulness is a requirement that cannot currently be tested. For the proposed taxonomy to be useful, the security community must see it as useful and use it in some way. It remains to be seen whether or not this taxonomy will be useful.

5.3 Future Work

The proposed taxonomy is a good start toward a taxonomy for computer and network attacks. In general it works well, and attacks are easily categorised. However, as always, there is room for improvement. As described in the above sections, some requirements have not been fully met and some areas could do with refinement.

A few problems were identified in Section 5.2. Blended attacks were sometimes difficult as they contained numerous sub-attacks. The issue here is not so much the taxonomy, but how the blended attacks have been analysed and described. Sometimes blended attacks are analysed in a way that mixes sub-attacks together. Therefore, the classifier must be able to sift through blended attack descriptions to find the information required. Future work in how to sift through attack descriptions would be helpful.

Attacks that have targets (or vulnerabilities) that require other targets are not fully modelled in the taxonomy. It would be useful in future versions of the taxonomy to be able to relate items within a dimension

better. Relating items so that an attack can have a combination of targets that are required, rather than a list of targets that have no relationship, would be useful.

To help understand classifications better, and to correlate attacks, some form of visualisation would be useful. Due to taxonomy having four dimensions, it is a non-trivial task. However, even if not all the information contained within the dimensions is presented, some form of visualisation allowing correlation between attacks would be helpful.

Research on how correlation between attacks within the taxonomy would be interesting. The dimensions allow for attacks to be correlated through properties such as the vulnerabilities the attacks use. This means attacks that previously may have appeared to have nothing in common, can be related through one of the dimensions. More research could be done on how this works and how beneficial it could be.

Further work could be done in moving the taxonomy toward a knowledge base approach. That is, as new classifications are created, they are added to a knowledge base. The knowledge base could detect correlations and allow for greater analysis of existing attacks. Another aspect would be the classification process. A step-by-step questionnaire could be used to ease classification. For example, the first few steps for classifying a worm in the first dimension might consist of:

- Is the attack self-replicating? (Yes = worm or virus, No = other 1st dimension attack)
- Does the self-replicating attack propagate through infected files? (Yes = virus, No = worm)
- Does the worm spread through email? (Yes = mass mailing worm, No = network aware worm)
- ...

This would continue until the worm has been classified in the all dimensions and would make the process of classifying easier and reduce the chance of error.

A more in-depth analysis of the taxonomy is required. While the above evaluation gives some idea of how well the taxonomy works, a more rigorous study should be conducted as future work. Further evaluation could include classifying a large number of attacks. If a knowledge base was implemented, artificial intelligence (AI) could be used to test the taxonomy using the knowledge base. The knowledge base could be learnt by the AI, then new attacks could be given to the AI to classify.

Chapter 6

Conclusion

Since the invention of computers and networks, people have found various ways to attack them. Attacks over the years have ranged from using a sledge hammer on a computer, to advanced distributed denial of service attacks. This research has focused on computer and network attacks and providing a taxonomy of them to help combat new attacks and to help computer and network security.

Before examining the two main areas of research, attacks and the taxonomy, a brief introduction to the field of computer and network attacks was given in Chapter 2. Previous work done in this field was discussed, as well as where the field is at currently, in regards to recent developments.

In Chapter 3, a wide range of computer and network attacks were discussed. This examination both helped to establish knowledge of attacks, which is helpful in combating attacks, as well as laying a foundation down for proposing a taxonomy. A taxonomy requires knowledge of the area being classified, thus examining the attacks was crucial.

The taxonomy was proposed in Chapter 4. Before proposing the taxonomy, existing taxonomies were examined and evaluated. Requirements for the taxonomy were also defined with the help of past research. The proposed taxonomy consists of four dimensions to provide a holistic approach to classifying attacks, and to deal with the problem of blended attacks. The first dimension covered the attack vector and the main behaviour of the attack. The second dimension allowed for classification of targets. Vulnerabilities were classified in the third dimension and payloads in the fourth.

The proposed taxonomy does not provide a description of attacks. For that level of detail classifying bodies, such as CERTs, provide detailed descriptions of attacks. Instead, the proposed taxonomy provides a classification which can be used to correlate similar attacks. This correlation allows for knowledge from older attacks to be applied to new attacks.

The taxonomy provides a common way of talking about attacks. Currently attacks are often described in different ways by different organisations. Some attacks are called viruses at one organisation, while another describes them as worms. The proposed taxonomy will remove this ambiguity by providing a common means of classifying the attacks.

In Chapter 5, the taxonomy was briefly examined and found to be, in general, a good way of classifying attacks. The requirements stated previously were re-examined and the taxonomy managed to meet most of them. A few improvements could be made which were suggested in Section 5.3.

A taxonomy allows for better understanding of attacks, and better understanding allows for better defence. Thus, the proposed taxonomy will benefit the security of networks and computers as it provides a more systematic way of understanding attacks.

Appendix A

Classifications of Case Studies

Table 5.2 shows the results of classifying a number of attacks using the proposed taxonomy. The table shows the first, second and fourth dimensions in full, but the second dimension has been truncated to show only the final entry. So for example, Code Red's second dimension is Software → Application → Server → Web → IIS → Versions 4, 5, and 6.0 beta, but only IIS 4, 5 and 6.0 beta is shown.

Also some entries are not complete, for example the Land attack has more than 40 different operating systems that it targets. Only a few of these are shown, but in a complete entry, all targets would be included.

Table A.1: Classification Results

Attack	1st Dimension	2nd Dimension	3rd Dimension	4th Dimension
Blaster[28]	Network-Aware Worm	MS Windows NT 4.0, 2000, XP, Server 2003	CAN-2003-0352	TCP packet flooding DoS
Chernobyl[20]	File Infector Virus	MS Windows 95 & 98		Corruption of Information
Code Red[23]	Network-Aware Worm	IIS 4, 5, & 6.0 beta	CVE-2001-0500	Stack Buffer Overflow & TCP packet flooding DoS.
John the Ripper ^a [63]	Guessing Password Attack	Unix Family, Windows NT, 2002 & XP	Configuration	Disclosure of Information
Infector[8]	File Infector Virus	DOS Family		Host Based Crasher DoS
Land[17]	Crasher DoS	Windows 95 and NT 4.0, Windows for Workgroups 3.11, ...	CVE-1999-0016	
Melissa[19]	Mass-Mailing Worm	Microsoft Word 97 & Word 2000	Configuration	Macro Virus & TCP packet flooding DoS
Michelangelo[15]	System/Boot Record Infector Virus	DOS Family		Corruption of Information
Nimda[24]	Mass-Mailing Worm	MS IE 5.5 SP1 & earlier except 5.01 SP2	CVE-2001-0333 & CVE-2001-0154	File Infector Virus, Trojan & DoS
PKZIP 3 Trojan[64]	Trojan	MS DOS		Corruption of Information
Ramen[25]	Network-Aware Worm	Redhat 6.2 & 7.0	CVE-2000-0573, CVE-2000-0666 & CVE-2000-0917	Host-based DoS, UDP & TCP packet flooding DoS & Subversion
Slammer[27]	Network-Aware Worm	MS SQL Server 2000	CAN-2002-0649	Stack Buffer Overflow & UDP packet flooding DoS
Sobig.F[30]	Mass-Mailing Worm	Email Client	Configuration	Trojan
Wuarchive FTPD ^b [16]	Trojan	Unix Family		Subversion

^aJohn the Ripper is a password cracking program.^bVersions 2.2 and 2.1f.

Bibliography

- [1] The Tribe Flood Network Distributed Denial of Service Attack Tool. October 1999. <http://staff.washington.edu/dittrich/misc/tfn.analysis>.
- [2] R. P. Abbott, J. S. Chin, J. E. Donnelley, W. L. Konigsford, S. Tokubo, and D. A. Webb. Security Analysis and Enhancements of Computer Operating Systems. Technical Report NBSIR 76 1041, Institute for Computer Sciences and Technology, National Bureau of Standards, April 1976.
- [3] admin@cgisecurity.com. The Cross Site Scripting FAQ. 2003. <http://www.cgisecurity.com/articles/xss-faq.shtml>.
- [4] Edward Amoroso. *Fundamentals of Computer Security Technology*. P T R Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- [5] Chris Anley. Advanced SQL Injection in SQL Server Applications. Technical report, NGSSoftware Insight Research (NISIR), 2002.
- [6] Ofir Arkin. ICMP Usage in Scanning: The Complete Know-How. *The Sys-Security Group*, 2001. http://www.sys-security.com/archive/papers/ICMP_Scanning_v3.0.pdf.
- [7] Taimur Aslam. A Taxonomy of Security Faults in the Unix Operating System. Master's thesis, Purdue University, 1995.
- [8] Network Associates. Infector virus. December 1992. http://vil.nai.com/vil/content/v_600.htm.
- [9] Hal Berghel. The Code Red Worm. *Communications of the ACM*, 44(12):15–19, 2001.
- [10] M. Bishop and D. Bailey. A critical analysis of vulnerability taxonomies, September 1996.
- [11] Matt Bishop. A Taxonomy of (Unix) System and Network Vulnerabilities. Technical Report CSE-9510, Department of Computer Science, University of California at Davis, May 1995.
- [12] Matt Bishop. Vulnerabilities Analysis. *International Symposium on Recent Advances in Intrusion Detection*, 1999.
- [13] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Seventh Annual International Conference on Mobile Computing And Networking*, 2001.
- [14] F Buchholz, T.E. Daniels, J.P. Early, R. Gopalakrishna, R.P. Gorman, B.A. Kuperman, S. Nystrom, A. Schroll, and A. Smith. Digging for Worms, Fishing for Answers. In *18th Annual Computer Security Applications Conference*, pages 219–226, 2002.
- [15] CERT Coordination Center. Advisory CA-1992-02 Michelangelo PC Virus Warning. February 1992. <http://www.cert.org/advisories/CA-1992-02.html>.
- [16] CERT Coordination Center. Advisory CA-1994-07 Wuarchive FTPD Trojan Horse. April 1994. <http://www.cert.org/advisories/CA-1994-07.html>.

- [17] CERT Coordination Center. Advisory CA-1997-28 IP Denial-of-Service Attacks. December 1997. <http://www.cert.org/advisories/CA-1997-28.html>.
- [18] CERT Coordination Center. Denial of Service Attacks. 1997. http://www.cert.org/tech_tips/denial_of_service.html.
- [19] CERT Coordination Center. Advisory CA-1999-04 Melissa Macro Virus. March 1999. <http://www.cert.org/advisories/CA-1999-04.html>.
- [20] CERT Coordination Center. Incident Note IN-99-03 CIH/Chernobyl Virus. April 1999. http://www.cert.org/incident_notes/IN-99-03.html.
- [21] CERT Coordination Center. Incident Note IN-99-07. November 1999. http://www.cert.org/incident_notes/IN-99-07.html.
- [22] CERT Coordination Center. Advisory CA-2001-13 Buffer Overflow in IIS Indexing Service DLL. June 2001. <http://www.cert.org/advisories/CA-2001-13.html>.
- [23] CERT Coordination Center. Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL. July 2001. <http://www.cert.org/advisories/CA-2001-19.html>.
- [24] CERT Coordination Center. Advisory CA-2001-26 Nimda Worm. September 2001. <http://www.cert.org/advisories/CA-2001-26.html>.
- [25] CERT Coordination Center. Incident Note IN-2001-01 Widespread Compromises via "ramen" Toolkit. January 2001. http://www.cert.org/incident_notes/IN-2001-01.html.
- [26] CERT Coordination Center. Trends in Denial of Service Attack Technology. 2001. http://www.cert.org/archive/pdf/DoS_trends.pdf.
- [27] CERT Coordination Center. Advisory CA-2003-04 MS-SQL Server Worm. January 2003. <http://www.cert.org/advisories/CA-2003-04.html>.
- [28] CERT Coordination Center. Advisory CA-2003-20 W32/Blaster Worm. August 2003. <http://www.cert.org/advisories/CA-2003-20.html>.
- [29] CERT Coordination Center. CERT/CC Statistics. 2003. http://www.cert.org/stats/cert_stats.html.
- [30] CERT Coordination Center. Incident Note IN-2003-03. August 2003. http://www.cert.org/incident_notes/IN-2003-03.html.
- [31] Xerox Palo Alto Research Center. Parc history. 2003. <http://www.parc.xerox.com/about/history/default.html>.
- [32] E. Chien and Peter Szor. Blended Attacks Exploits, Vulnerabilities and Buffer-Overflow Techniques in Computer Viruses. *Virus Bulletin Conference*, 2002.
- [33] Fred Cohen. *Computer Viruses*. PhD thesis, University of Southern California, 1985.
- [34] CVE. Common Vulnerabilities and Exposures. 2003. <http://www.cve.mitre.org/>.
- [35] Marco de Vivo, Gabriela O. de Vivo, and Germinal Isern. Internet Security Attacks at the Basic Levels. *ACM SIGOPS Operating Systems Review*, 32(2):4–15, 1998.
- [36] David Dittrich. The DoS Project's "trinoo" Distributed Denial of Service Attack Tool. October 1999. <http://staff.washington.edu/dittrich/misc/trinoo.analysis>.
- [37] David Dittrich. The "stacheldraht" Distributed Denial of Service Attack Tool. December 1999. <http://staff.washington.edu/dittrich/misc/stacheldraht.analysis>.

- [38] Mark Eichin and Jon Rochlis. With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988. Technical report, Massachusetts Institute of Technology, 1988.
- [39] Pierre-Alain Fayolle and Vincent Glaume. A Buffer Overflow Study: Attacks & Defenses. Technical report, Ecole Nationale Supérieure d'Electronique, Informatique et Radiocommunications de Bordeaux, 2002.
- [40] Scott Fluhrer, Itsik Mantin, and Adi Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. *Lecture Notes in Computer Science*, 2259, 2001.
- [41] L. Garber. Melissa Virus Creates a New Type of Threat. *Computer*, 32(6):16–19, 1999.
- [42] Samuel Glasstone and Philip J. Dolan. The Effects of Nuclear Weapons. *US AEC*, 1962, Revised Edition 1977.
- [43] A. Householder, K. Houle, and C. Dougherty. Computer Attack Trends Challenge Internet Security. *IEEE Computer*, 35(4):5–7, April 2002.
- [44] John D. Howard. *An Analysis Of Security Incidents On The Internet 1989-1995*. PhD thesis, Carnegie Mellon University, 1997.
- [45] John D. Howard and Thomas A. Longstaff. A Common Language for Computer Security Incidents. Technical report, Sandia National Laboratories, 1998.
- [46] Howard F. Lipson. Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues. Technical report, CERT Coordination Center, November 2002.
- [47] Richard Bisbey II and Dennis Hollingworth. Protection Analysis: Final Report. Technical report, University of Southern California, May 1978.
- [48] Insecure.org. Nmap: Network Mapper. 2003. <http://www.insecure.org/nmap/>.
- [49] Malachi Kenney. The Ping o' Death page. 1997. <http://flowserv.teco.uni-karlsruhe.de/ping/>.
- [50] Daniel V. Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. In *UNIX Security Workshop II*, August 1990.
- [51] Carlo Kopp. The Electromagnetic Bomb - a Weapon of Electrical Mass Destruction. *Aerospace Power Chronicles*. <http://www.airpower.maxwell.af.mil/airchronicles/kopp/apjemp.html>.
- [52] Ivan Victor Krsul. *Software Vulnerability Analysis*. PhD thesis, Purdue University, 1998.
- [53] ICSA Labs. Melissa Costs. 1999. <http://www.icsalabs.com/html/communities/antivirus/melissa/melissa8a.sh%tml>.
- [54] SPI Labs. SQL Injection: Are Your Web Applications Vulnerable? Technical report, SPI Dynamics, 2002.
- [55] Ulf Lindqvist and Erland Jonsson. How to Systematically Classify Computer Security Intrusions. *IEEE Security and Privacy*, pages 154–163, 1997.
- [56] Daniel Lowry Lough. *A Taxonomy of Computer Attacks with Applications to Wireless Networks*. PhD thesis, Virginia Polytechnic Institute and State University, 2001.
- [57] Joe Loughry and David A. Umphress. Information Leakage from Optical Emanations. *ACM Transactions on Information and System Security (TISSEC)*, 5(3):262–289, 2002.
- [58] David E. Mann and Steven M. Christey. Common Vulnerabilities and Exposures. Technical report, The MITRE Corporation, 1999. <http://www.cve.mitre.org/docs/cerias.html/>.

- [59] Fauzan Mirza. Word for Windows Password Cracker (R8). 1995. <http://snake.cs.tu-berlin.de:8081/~schwartz/pmh/elser/contrib/wfwcd.zip>.
- [60] Cult of the Dead Cow. Back Orifice 2000. 2003. <http://www.bo2k.com>.
- [61] Cult of the Swimming Elephant. Password Tools. 2003. <http://www.cotse.com/tools/password.htm>.
- [62] Airsnort Project. Airsnort. 2003. <http://airsnort.shmoo.com/>.
- [63] Openwall Project. John the Ripper Password Cracker. 2003. <http://www.openwall.com/john/>.
- [64] Chris Rodgers. Threats to TCP/IP Network Security. 2001.
- [65] McAfee Security. Virus Information Library. 2003. <http://www.mcafee.com/anti-virus/default.asp>.
- [66] SecurityFocus.com. BugTraq. <http://www.securityfocus.com>.
- [67] Eugene Spafford. The Internet Worm Program: An Analysis. Technical report, Department of Computer Sciences, Purdue University, 1988.
- [68] S.R. Subramanya and N. Lakshminarasimhan. Computer Viruses. *IEEE Potentials*, 20(4):16–19, Oct/Nov 2001.
- [69] Symantec. Symantec Internet Security Threat Report Volume III. February 2003. <http://enterprisesecurity.symantec.com/content.cfm?articleid=1539&EID=0>.
- [70] Symantec. Symantec Internet Security Threat Report Volume IV. September 2003. <http://enterprisesecurity.symantec.com/content.cfm?articleid=1539&EID=0>.
- [71] Internet Security Systems. ISS Security Advisory: "Snork" Denial of Service Attack Against Windows NT RPC Service. 1998. <http://xforce.iss.net/xforce/alerts/id/advis9>.
- [72] OASIS WAS TC. OASIS Web Application Security Technical Committee. 2003. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=was.
- [73] OASIS WAS TC. OASIS Web Application Security Technical Committee list archives. 2003. <http://lists.oasis-open.org/archives/was/>.
- [74] Wim van Eck. Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk? *Computers & Security*, 4(4), December 1985.
- [75] Cliff Changchun Zou, Weibo Gong, and Don Towsley. Code Red Worm Propagation Modeling and Analysis. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 138–147. ACM Press, 2002.